

## Introduction

This document presents a recommendation to redesign and expand the existing Twitch Developer Portal, dev.twitch.tv. The Twitch Developer Portal (TDP) is a key component of Twitch and Amazon's initiative to support and engage with game makers who want to optimize their game experiences for Twitch. The TDP's vision is to help everyone on Earth deliver the next generation of entertainment on Twitch. This recommendation includes a design direction for the TDP based on 13 weeks of customer research, design iterations, and usability testing. This document includes background information on the existing developer portal, a project overview, the early design approach, and a detailed design recommendation for the next phase.

## Background

Amazon has invested over \$1B in games with Amazon Game Studios, Amazon gaming platforms, and now Twitch<sup>1</sup>. Twitch is the largest video game streaming site in the world, and in 2014 it became the largest subsidiary acquisition in Amazon's history. During peak hours Twitch commands 2% of all Internet traffic in the US and accounts for 40% of all live-streamed content. Amazon bought Twitch to leverage the 100mm monthly millennial active users on Twitch<sup>2</sup>; of which only 20% have Amazon accounts<sup>3</sup>.

The Amazon Twitch Seattle team has been tasked with defining the next generation of Twitch experiences, even as competition from other companies such as Steam, YouTube, Afreeca, and Niconico intensifies. At the 2015 Game Developers Conference and Electronic Entertainment Expo, the Twitch Seattle team gauged interest in customizing game experiences for Twitch broadcasters by talking to over 20 games developers from Activision/Sledgehammer Studios, Blizzard, Beatshapers, Bethesda, Insomniac, Industrial Toys, Wargaming, Warner Brothers, GLU, Monolith, and Kabam; all of whom want to engage with Twitch but don't know how<sup>1</sup>. Up until now, Twitch's main priority has been growing its user base by supporting the broadcasters who stream games live; Twitch has not yet built strong mutually beneficial relationships with game developers. Today, Twitch sees engaging with game creators as a key step to revolutionizing the entertainment industry. To support this initiative, the TDP will revitalize the dev.twitch.tv site and help developers create games for Twitch by supplying developer support tools, technical documentation, and help articles. The TDP project, initially spearheaded by Amazon, will be handed to the Twitch design team in August 2015 for continued exploration and implementation<sup>1</sup>. This project signals Twitch's strong interest in partnering with game creators to optimize and extend their games.

## Project Overview

Our goal is to help the world create next-generation entertainment experiences and we believe that supporting game creators is the best way to do so. We have identified the TDP as a key component that game creators can leverage to envision these new experiences and make them a reality. The Twitch Seattle team tasked with this project evaluated the project timeline and determined the need for a two-pronged design direction. First, the current TDP needs an immediate revamp for TwitchCon in September 2015. A focused design, targeted at experienced AAA game developers, will be released in time for several critical TwitchCon announcements. Second, a broader portal design, scaled to reach all game creators, marketers, and broadcasters, will be released in early 2016 to support the Electronic Entertainment Expo.

The design approach for this project is divided into three phases, with each producing a recommendation for both a focused and scalable design. The first design phase, early design, defines the TDP's key customer base and validates the design direction through research. The second design phase, detailed design, will refine the early design mockups. The final design phase will be implementation.

---

<sup>1</sup> Interview with Abizar Vakharia, Twitch Seattle Senior Product Manager

<sup>2</sup> <http://www.businessinsider.com/amazon-buys-twitch-2014-8>

<sup>3</sup> Cross reference of accounts between orgs

As the early design project owner, I launched this assignment with the assistance of an advisory team, consisting of stakeholders in Seattle and at Twitch. I worked closely with Twitch throughout the early design process by continuously presenting and defending my designs. Over the course of 13 weeks, I conducted preliminary research, design iterations, and usability testing. The early designs will facilitate the detailed design phase continued by Twitch.

## Early Design Approach

### Preliminary Research

To determine the design requirements for this project, two phases of preliminary research were conducted. The first phase consisted of interviews with potential TDP customers. Throughout the week of 4/14, we interviewed four senior game developers, three junior-level game and software developers, two producers, and one designer, all from within Amazon Game Studios. These interviews helped determine the key function a developer portal serves for each of these game creator roles (see Appendix B for user stories). The next research phase consisted of competitive analysis of other open-access developer portals. We identified seven competitors and documented each competitor's core features (see Appendix B for competitive analysis).

The preliminary research validated our assumption that our key customers are game developers, designers, and producers. Game marketers, Twitch broadcasters, and plugin or app developers are additional TDP customers that were identified in the next research phase, usability testing. To accommodate the urgency of the initial focused redesign, it was decided that this effort will be targeted at experienced AAA game developers only. The second scalable design will encompass all the key customers: game developers, designers, producers, marketers, broadcasters, and plugin/app developers. Experienced AAA game developers were chosen as the initial target because these customers want to integrate their game with Twitch in order to promote the discovery of their AAA title, and they have the technical know-how to do so.

### Design Requirements

For the target user, the most important requirements are (1) code samples (2) search (both internal and external) and (3) tutorials. Additional requirements include developer guides, reference documentation, tools/article pages, Help/FAQ centers, visuals, and news & release cadences. (See Appendix B)

### Early Concept Designs

The design requirements were leveraged to create site maps for both the initial focused design and scalable design. The site maps were mocked up (see Appendix C) and presented to Twitch 7/14. Feedback from Twitch validated the design direction and moved the project into the usability-testing phase.

### Usability Testing

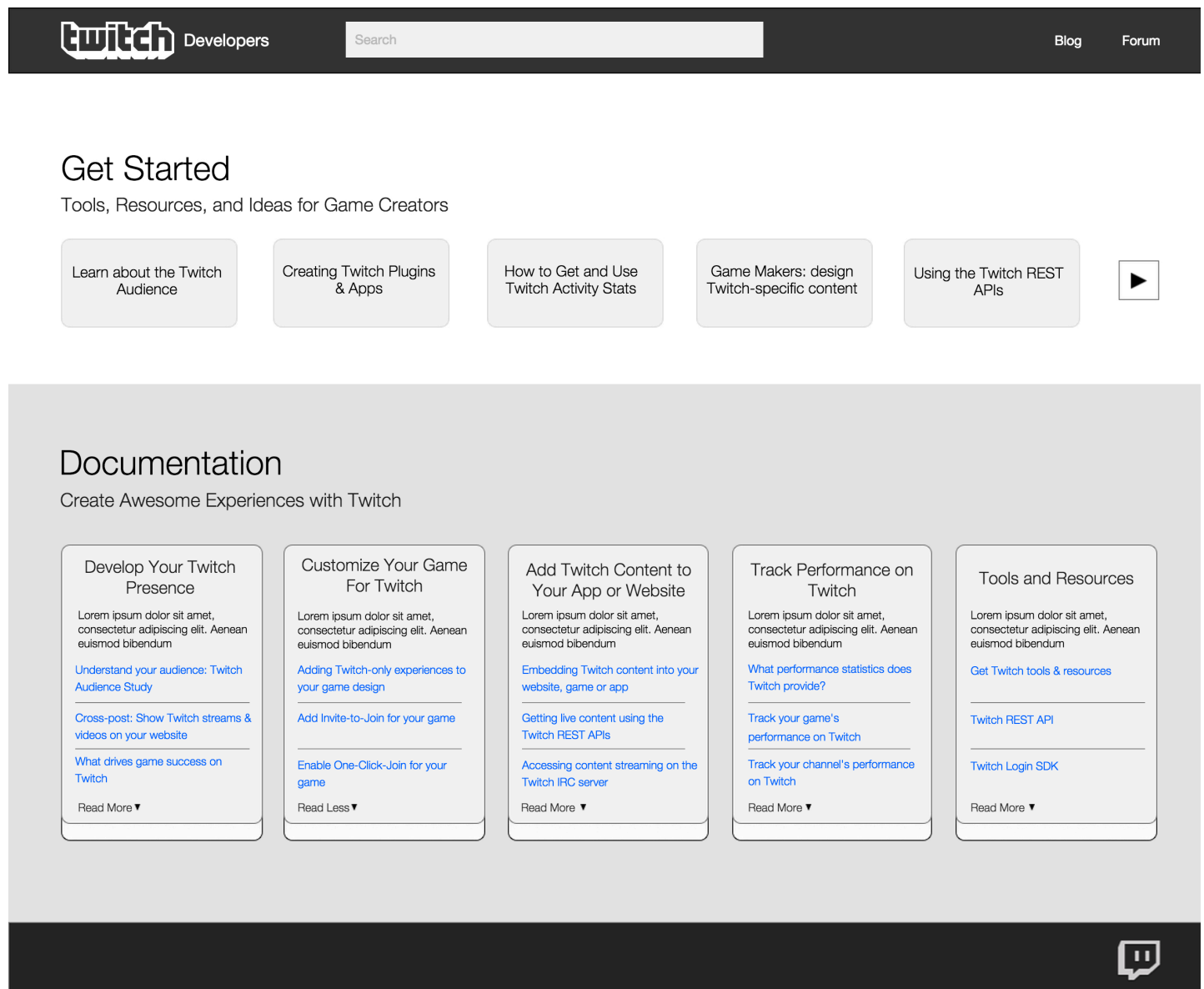
A usability study was conducted to validate the design direction of the TDP and the site information architecture with key customers. The study was conducted over seven days starting 7/24. A total of 20 participants were recruited, consisting of 4 game producers, 2 junior-level developers, 7 senior-level game developers, 3 game designers, 2 Twitch broadcasters, and 2 game marketers. Each participant completed a card sort test in which he or she was given a list of topics and asked to sort them by relevance and group them into participant-generated categories. The card sort provided data on how participants expected TDP content to be organized and on whether our structure was easily interpreted. A select number of participants were also asked to complete a task-based scenario using an interactive prototype. A moderator recorded each participant's feedback for analysis later. For detailed methodology and data findings, see Appendix D.

The key insights from the test found that: (1) participants tended to group topics by task to complete; (2) game developers are only interested in technical topics; (3) game developers are answer-driven, while producers, designers, and marketers are more interested in browsing; (4) participants were unclear of their next steps after landing on the Home Page; (5) producers want reliable information about product status and future releases; and (6) participants were confused by the information structure on the Article Pages, specifically the relationship between the content and topic navigation. For details and logic, see Appendix D.

## Recommendations

Based on the usability study findings, I recommend the following set of actionable requirements for the early design. (1) Use an information architecture that follows a “task-driven” model rather than a role-based model. (2) Provide a role filter feature that enables users to quickly find topics most relevant to them. (3) Include both browsing and search to accommodate all users’ preferences. (4) Provide a clear call to action on the Home Page. (3) Create a timeline of product news & releases to include both recent updates and future plans. (6) Apply some key best practices for visually grouping information on the Article Pages. The Home Page design recommendation for both the focused and scalable design is provided below. Appendix E contains the final early design spec, which includes design logic for both the focused design and the scalable design.

### Focused Home Page Design

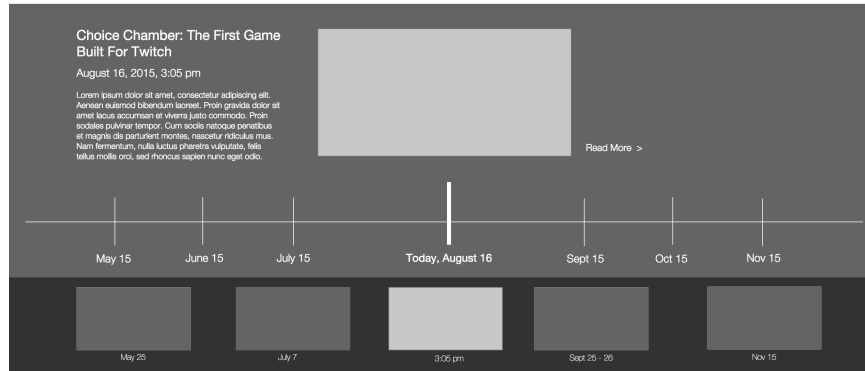


# Scalable Home Page Design



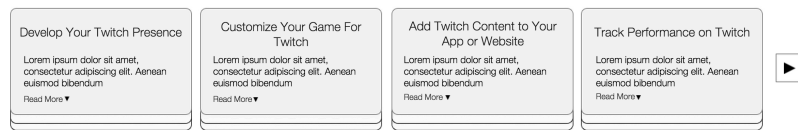
## Get Started

Tools, Resources, and Ideas for Game Creators



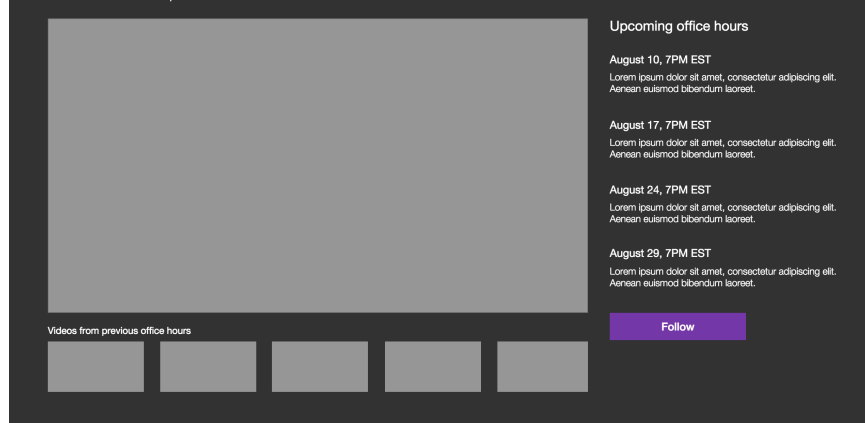
## Documentation

Create Awesome Experiences with Twitch



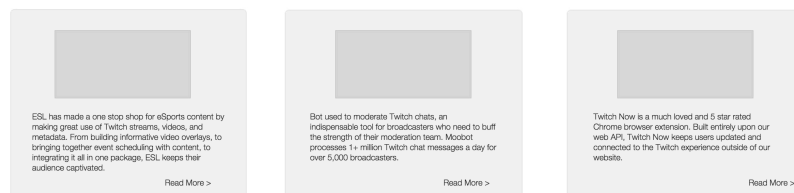
## Watch & Learn

Twitch Game Development Office Hours



## Success Stories

Games Exclusively for the Twitch Audience





## FAQ

### When will the TDP launch?

The first iteration of the TDP redesign will launch at TwitchCon in September 2016. This design is targeted at experienced AAA game developers. In early 2016, the second iteration of the TDP will be launched for all game creators including: game developers, designers, producers, marketers, and broadcasters.

### Who will own this project?

Twitch will own this project starting August 2015. The owner of these pages will likely be a combination of product marketing, business development, and design<sup>4</sup>.

### Is there any content currently on dev.twitch.tv? Will it be incorporated into the TDP?

Yes, the current site links to the Twitch GitHub repository, and also highlights several companies using Twitch. Twitch wants to continue promoting the APIs and what can be built using the Twitch REST APIs. The API and success story content will be brought into the redesign as documentation articles and home page success articles.

### When was dev.twitch.tv initially launched?

Twitch launched dev.twitch.tv in March 2014. The initial purpose of the site was to promote the Twitch Broadcasting SDK to various game publishers, and required integrating games on a one-on-one basis, which proved to be costly. Several months later, Sony and Microsoft began integrating broadcasting on a system level to stream directly with Twitch. These efforts rendered the Broadcasting SDK obsolete, and dev.twitch.tv has sat largely unused for over a year<sup>4</sup>.

### Have you looked at other competitor developer portals? What content do they offer?

We conducted an analysis to better understand the competitive landscape but chose not to fixate on our competitors. Competitive analysis focused on open-access developer portals that corresponded with a product or service, such as an API or SDK. We analyzed the core features of each competitor and found that all sites incorporated documentation and code samples. We also identified that only 57% of the portals incorporated search and tutorials, which are important features to our target users. Read more in Appendix B.

### What are the primary goals of the key users coming to the TDP?

Appendix B provides a full list of TDP key user stories. We identified four primary user stories that helped direct our design iterations: (1) new visitors arrive to the site via a specific topic page and then navigate to the home page; (2) experienced game developers scan sample code for information; (3) game designers need to understand the different tools available on Twitch; and (4) producers want to find Twitch software release cadence information.

### How did you select participants for the usability study?

Over the course of a week we interviewed 20 participants within AGS and Twitch. Prior to the study we identified two requirements: (1) each participant must be a senior-level game maker, and (2) there must be at least two participants from each game-maker category. For a full participant profile breakdown see Appendix D.

### What topic categories are you proposing?

The usability participants identified three distinct categories of topics in the card sort categorization test. We considered the groupings and adjusted the categories to reflect the content topic. Our goal is to align our categories with user expectations. A full list of the topics and categories can be found in Appendix D.

---

<sup>4</sup> Interview with David McLeod, Twitch Design Director

## Appendix A: Press Release

### Press Release: Twitch Challenges the World to Create New Entertainment Experiences *Game creators rise to new heights with the help of dev.twitch.tv*

March 14, 2016 (San Francisco) – Today Twitch announced the launch of their new Developer Portal, dev.twitch.tv. With this new resource to aid them, game creators are no longer stymied on how to create great new game features exclusively for use in Twitch stream. This and other types of next-generation game experience is widely seen by game publishers as the golden key to winning coveted visibility and engagement with can now optimize and extend their games for the Twitch platform and engage Twitch’s 100mm (and growing) monthly active users. The Developer Portal will provide all game creators including developers, designers, producers, and marketers as well as Twitch broadcasters will the tools and resources that will spark their creative imagination. Game makers will find one place for all information on what’s possible to do on Twitch and how to ingrate and extent their games for Twitch.

“This resource makes it possible to add 3-4 new Twitch features to our next compendium. Stay tuned!” – Blizzard Senior Producer

“I was part of the Developer Library’s initial research group and was excited about what I saw. I just green lit a new game designed around Twitch audience participation.” – Insomniac Director of Engineering

This new portal is a response to requests from game creators who want to reach the huge gaming audience Twitch offers. Game experiences produced exclusively for Twitch have higher visibility then other games on Twitch because broadcasters and viewers can engage with the game like never before. Game developers that have already produced Twitch experiences have seen 40% higher customer conversion from watchers to players compared to other top games.

“We are getting ready to launch our new game Holiday 2016 and this announcement took us back to the drawing board. This resource made us rethink our game promotion strategy. We now have the tools to rely on Twitch as our primary discovery tool.” – Bethesda CMO

The Twitch Developer Portal is an exciting step forward for the game creator community. All game tools and resources are available at dev.twitch.tv.

### TDP Key User Stories

I find out about dev.twitch.tv from online news articles and word of mouth.

I use Google to search a specific topic and navigate to the Twitch developer portal.

As a new visitor I arrive at the site via a specific topic page then navigate to the home page.

As an experienced game developer, I scan sample code to find the information I need.

As a new visitor I have confidence I found the right information in my search because I'm able to review the most popular features Twitch provides and see the high-level navigation on the landing page.

As a producer I visit the home page to find information on Twitch software release cadences.

As a designer I visit the home page to understand the different topics and tools Twitch provides and to read article pages about new features.

I explore the site by scanning documentation to find subtopics relevant to features I'm designing.

As a junior developer I use the QA forum whenever I have a question that can't be answered by a quick scan of the documentation.

As a return visitor I navigate to the Twitch developer blog to stay informed on new announcements.

### TDP Customer Scenario

An AAA game studio wants to compete in the in-app purchase market for PC games. The game producer meets with the leaders of this studio and decides to make this PC game available on all PC game marketplaces. Additionally, the studio wants to enable more player engagement with the game through platform and marketplace integration. These business requirements are handed down to the game design team, which decides to create this game for the Twitch platform, among others. These designers visit the Twitch developer portal to see the integration tools available and use these insights to write the design requirements. The design team then hands off the requirements to the developer team. A group of experienced developers meet to determine if these design requirements fit the technical requirements of the Twitch platform. These developers visit the Twitch portal to scan the technical capabilities of the platform. Once each requirement is green-lit, the development management team assigns feature development to a junior developer. This junior developer is responsible for the implementation of each feature assigned to him or her. In order to start development, each junior developer conducts in-depth research on each feature by reading documentation and code samples on the Twitch developer portal. Throughout the development process the junior developer will return to the developer portal as a reference

### Twitch Competitive Analysis

Competitive analysis focused on open-access or open-source developer portals that corresponded with a product or service, such as an API or SDK. The goal of the TDP is to provide information on the integration of Twitch into games. We chose not to focus on platform developer portals such as Steamworks, Desura, Android, or IOS because these portals tend to be closed access and describe development platforms.

#### 1. Twitch Competitors

##### a. [Twilio Documentation](#)

- i. Twilio is an open-access portal focused on integrating Twilio cloud communication APIs with customer apps.

##### b. [Parse Documentation](#)

- i. Parse is an open-access portal targeting the download and use of Parse cloud hosting APIs for all languages and platforms.

##### c. [Stripe Documentation](#)

- i. Stripe is an open-access mobile payments portal targeted at answering questions via a documentation format.

##### d. [Zendesk Developers](#)

- i. Zendesk is an open-access portal focused on integrating Zendesk customer service APIs, apps, and SDK with customer apps.

##### e. [Facebook Developers](#)

- i. Facebook is an open-access portal focused on the integration of Facebook with customer apps.
- f. [Google Maps](#)
  - i. Google Maps is an open-access portal, which is part of the closed-access Google developer portal. Google Maps targets integrating Maps with customer apps.
- g. [Unity Documentation](#)
  - i. Unity is an open-access portal designed to provide documentation for developers using the platform.

## 2. Competitor Core Features

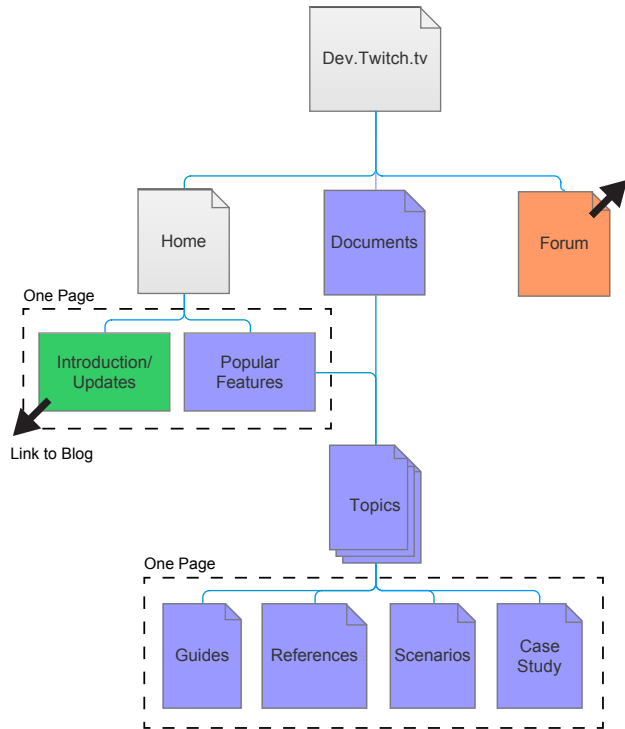
- a. The highlighted core competitor features are the features important to the target user: experienced AAA game developers.

	Docs	Code Samples	Search	Tutorial	Ref	Guides	SDK↓	Tools	News	Release Cadence	Help
Twilio	✓	✓		✓	✓	✓					✓
Parse	✓	✓			✓	✓					
Stripe	✓	✓									✓
Zendesk		✓	✓			✓	✓				✓
Facebook	✓	✓	✓			✓	✓	✓	✓	✓	✓
Maps	✓	✓	✓	✓	✓	✓				✓	✓
Oculus	✓	✓		✓		✓	✓		✓	✓	
Unity	✓	✓	✓								

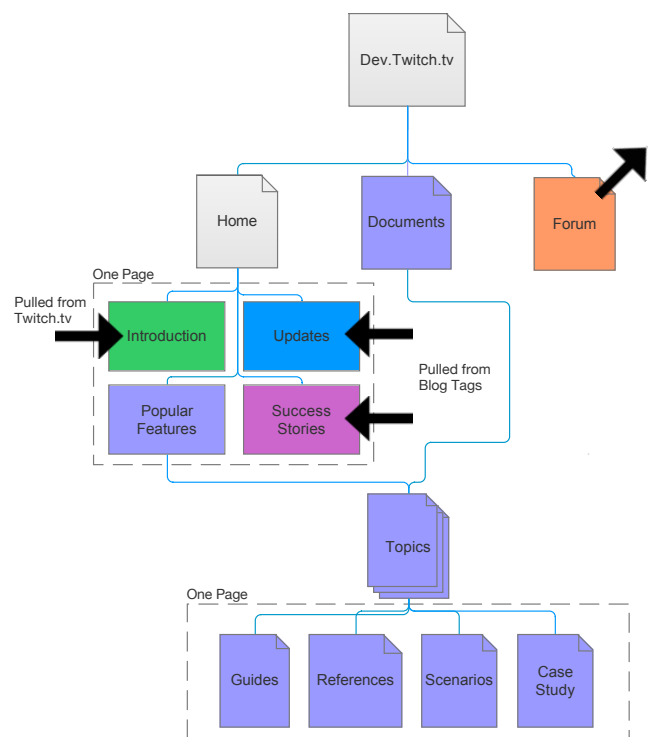
## Appendix C – Early Concept Designs

The focused site map reflects the needs of the customer: experienced AAA game developers. The Scalable site map reflects the needs of the target customer as well as producers, designers, marketers, and broadcasters.

Focused Site Map



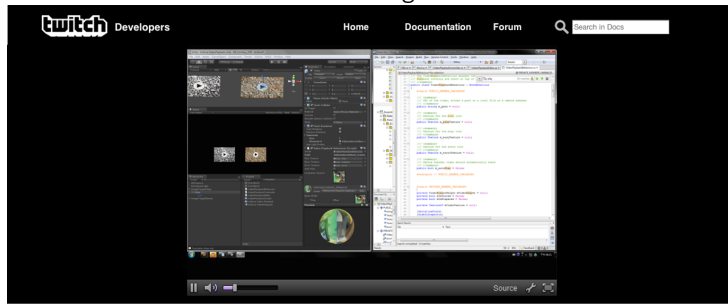
Scalable Site Map



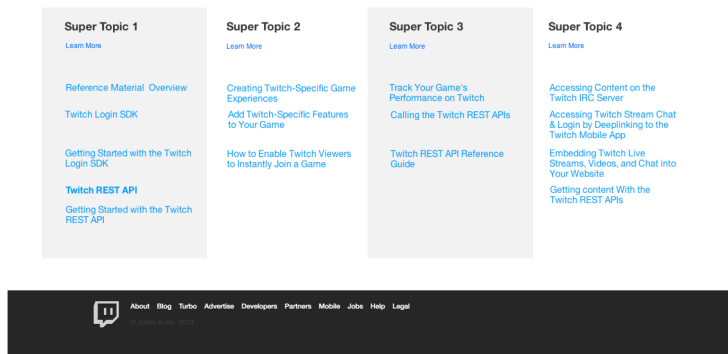
## Early Concept Designs Presented to Twitch 7/14

### Targeted Design

#### Home Page



#### Popular Twitch Features



Global Navigation: Home, Documentation, Forum, Search

Game development video feed pulled from Twitch.tv

Links to all documentation content. Mitigates need for a documentation landing page.

#### Article Page

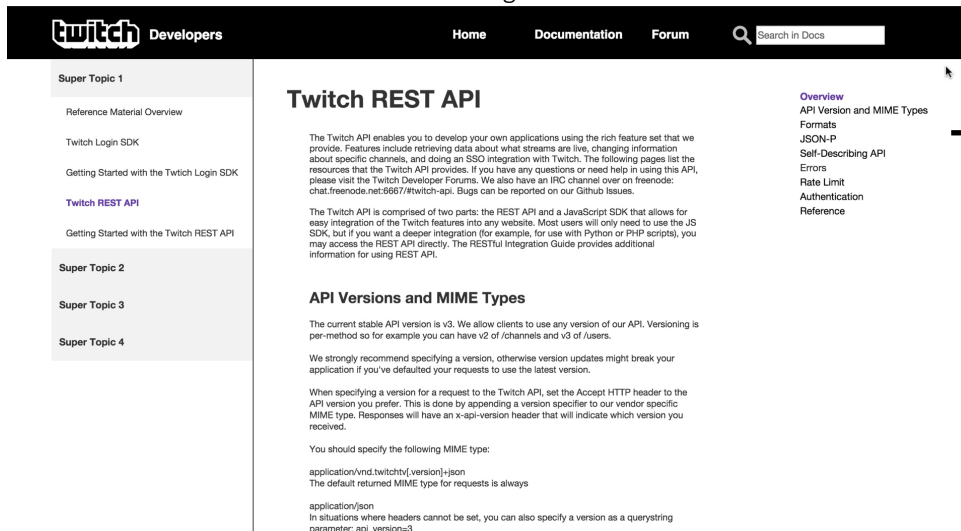
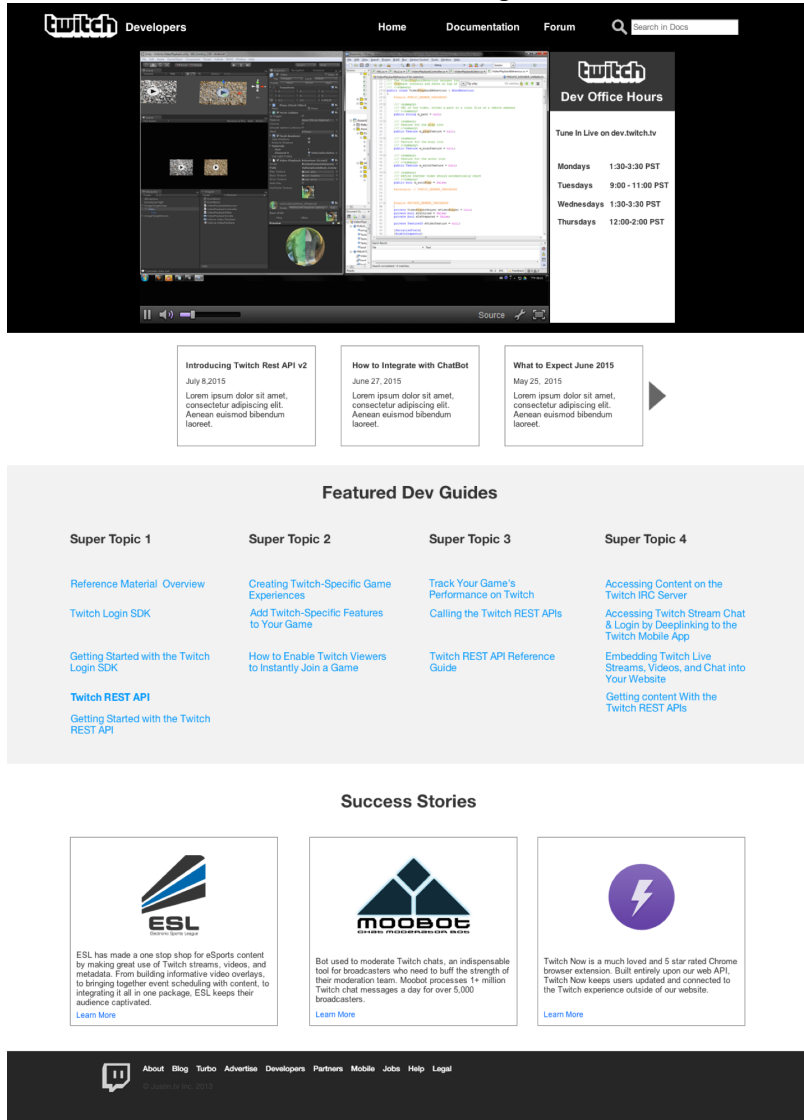


Table of contents.  
Dynamically updates position as user scrolls page

Left navigation sorted by categories. Each topic has its own content page.

## Home Page

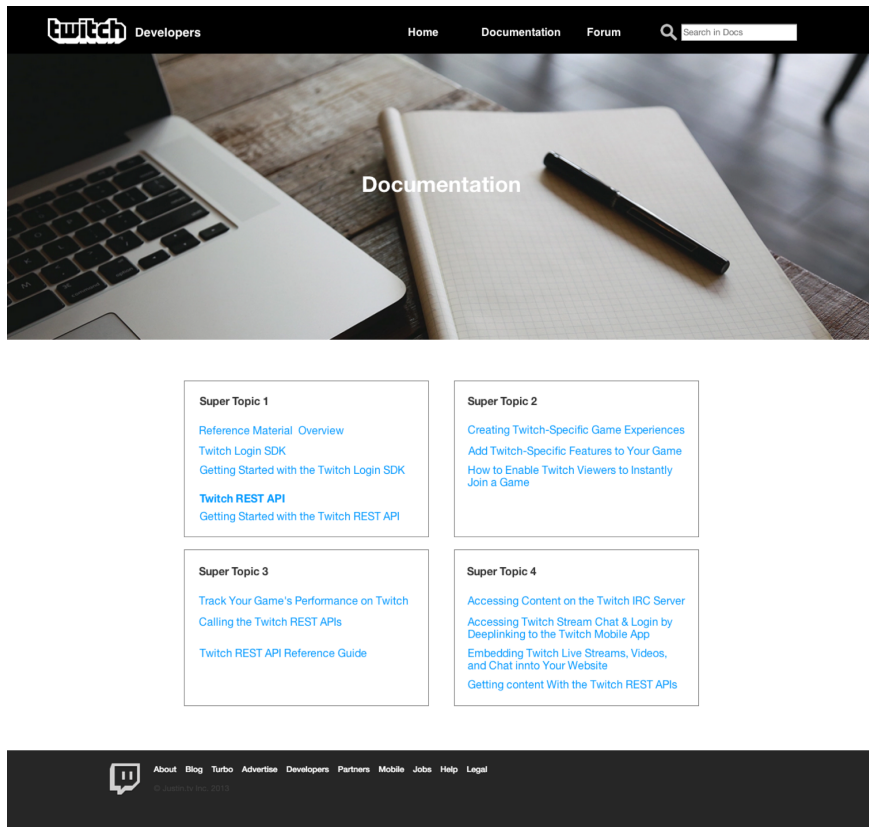


Recent updates pulled from blog.  
Targeted at Producers

Topics with the largest # of hits.  
Doesn't include all topics.

Success stories from current site.  
Targeted at marketers.

## Documentation Landing Page



Accessible through the documentation link in global navigation.

Topics grouped by category. Clicking any topic will take customer to documentation topic page.



### Research Questions

1. Should information be categorized by scenario/use case or by game-design role?
2. Do customers prefer a Topic Page layout with long scrollable pages or do customers prefer topics to be chunked into smaller pages?
3. Does the current design support quick access to content and minimize time spent searching?
4. Will customers find content pulled from the Twitch blog and forum helpful?
5. Does the Home Page design fulfill the requirements of producers, designers, and marketers?

### Methodology

- Card sorting- prior to the usability testing sessions, all 20 participants were sent a card sort test, which they were to complete. Participants spent an average of 20 minutes completing the test. The test consisted of two parts: a relevancy sort and a categorization sort. The first sort asked participants to indicate if each supplied topic was either relevant or irrelevant to his or her game-making role. The second sort required participants to group all the topics into categories of their choosing. The participants were also asked to name each category.
- Usability testing- Ten of the twenty participants were invited to a one-on-one UX interview session lasting 30 minutes. Each participant was asked their reasoning behind the sort and their motives for the categorization. Next the participants were shown an interactive prototype and given a scenario to complete using the prototype. A moderator recorded each participant's comments during the test.

### Recruiting criteria

- Must be inside the AGS and Twitch teams
- Must be a senior-level game maker with many years of industry experience
  - Excludes junior developer category
- At least two participants from each game-maker category

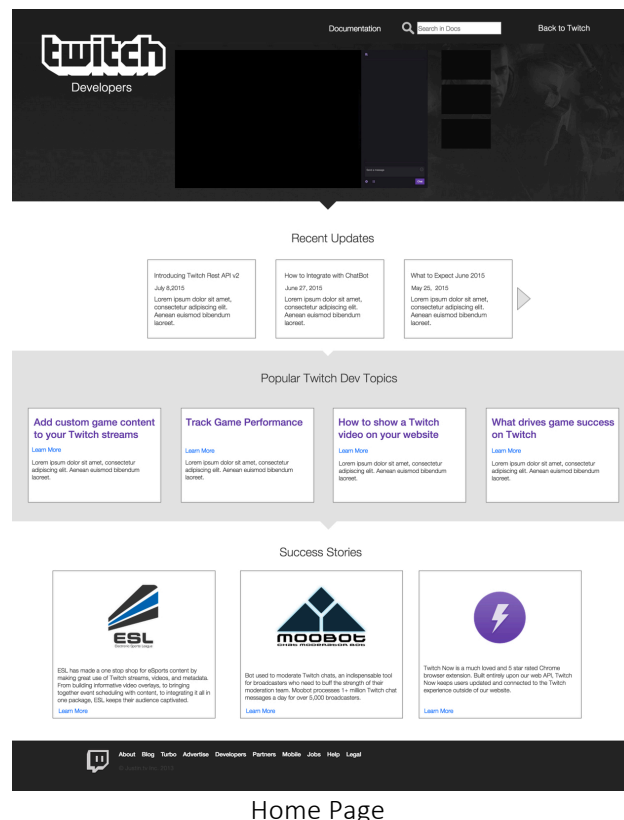
### Participant Profile

- 20 participants (18 Amazonians, 2 Twitch)
- 18 males, 2 females
- 4 game producers
- 2 junior-level developers
- 7 senior-level game developers
- 3 game designers
- 2 Twitch broadcasters
- 2 game marketers

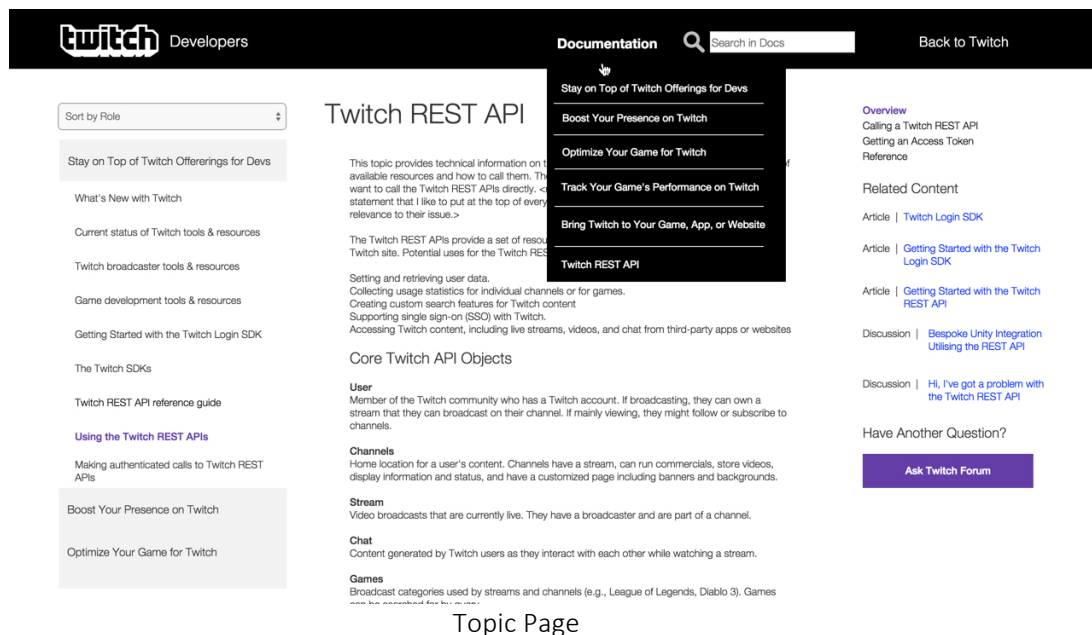
### Detailed Findings

1. Data supports “scenario-driven” information architecture
  - a. The card sort categorization data (outlined in Testing Data below) revealed that 40% of participants sorted topics by role, 40% sorted by scenario, and 20% sorted as a hybrid between the two. However, when looking at a 73% correlation between the data the three biggest user-generated categories are: Boost your presence on Twitch, Customize your game for Twitch, and Engage with the Twitch API & SDK. The first two categories are scenario driven while the third is role driven for game developers.
  - b. Additionally when participants were asked to complete a task using the interactive prototype, each participant, including game developers, chose to search or browse the prototype scanning for key words from the scenario.
2. Relevancy card sort supports a game-maker role filter system
  - a. According to the data, a scenario-driven information architecture works for first time visitors that prefer to search for an answer by their use case, not by their game-making role. However, according to the relevancy card sort, game developers are very particular regarding the information on a TDL that interests them.

- b. Referring to the relevancy card sort data, there was no overlap between the topics relevant to game developers and other game maker categories. Additionally in the usability prototype test, game developers expressed a strong interest in sorting information by their role once they landed on a Topic Page. Other game maker roles also expressed interest in sorting information by role, yet the primary users of this sort system would be game developers.
  3. Game developers are answer-driven, while producers, designers, and marketers are more interested in browsing
    - a. Developers are answer-driven and impatient searching for information. The usability prototype sessions affirmed our assumptions that developers aren't interested in browsing. In each of the of the three sessions conducted with developers participants immediately went to the internal search bar on the Topic Page or did control F to find the exact key word they were interested in.
    - b. Quote from participant: "I immediately want to do a control F to find the reference material. I don't like scrolling or flipping through content."
    - c. The usability prototype sessions confirmed our initial assumption that the Home Page would be targeted at first time visitors and producers, designers, and marketers. The Home Page was designed for browsing and the producers, designers, and marketers we talked to indicated an interest in browsing the Home Page as well as Topic Pages. Compared to developers, these users were inclined to click around the navigation on the Topic Pages to discover other content.
    - d. Quote from participant: "On a scale of 1 to 10 I'm most interested in browsing as a pleasure read or a weekend read. During the week I'm much less interested in browsing and searching for an answer quickly."
  4. Home Page needs to provide a clear call to action.
    - a. All usability prototype session participants arrived to the site via the Home Page. Once arriving about half of the participants were unclear of their next steps. These participants immediately started scrolling the page and weren't able to find what they were looking for in either the Popular Features tab nor anywhere else on the page. The other half of the participants either found the correct information from the Popular Features tab or through the documentation dropdown list in the global navigation.
    - b. Quote from participant: "I'm not entirely sure what to do next. I think there should be a Getting Started guide somewhere on the page."



5. Producers want a timeline of product and feature releases
  - a. As previously outlined, the Home Page was designed primarily for first time visitors, producers, designers, and marketers. The usability prototype Home Page was designed with a Recent Updates tab specially targeted at producers. Our initial research suggested producers needed this information to make development decisions and timelines.
  - b. The usability prototype sessions confirmed that producers want recent updates but they are more interested in product timelines and feature release cadences. Producers want to know what releases will be coming out in the next three to six months.
6. Article pages need visual groupings
  - a. The usability prototypes were designed with monochromatic Topic Pages. Participants were generally confused once navigating to these pages as to their location on the site. Participants were also initially unfamiliar with how the navigation worked, particularly how information was grouped. We believe visually grouping the major sections of the page: navigation, content, and related links will help the customer easily find information in the navigation and understand their location on the site.



Topic Page

## Testing Data

### Card Sort Relevancy Data by Role

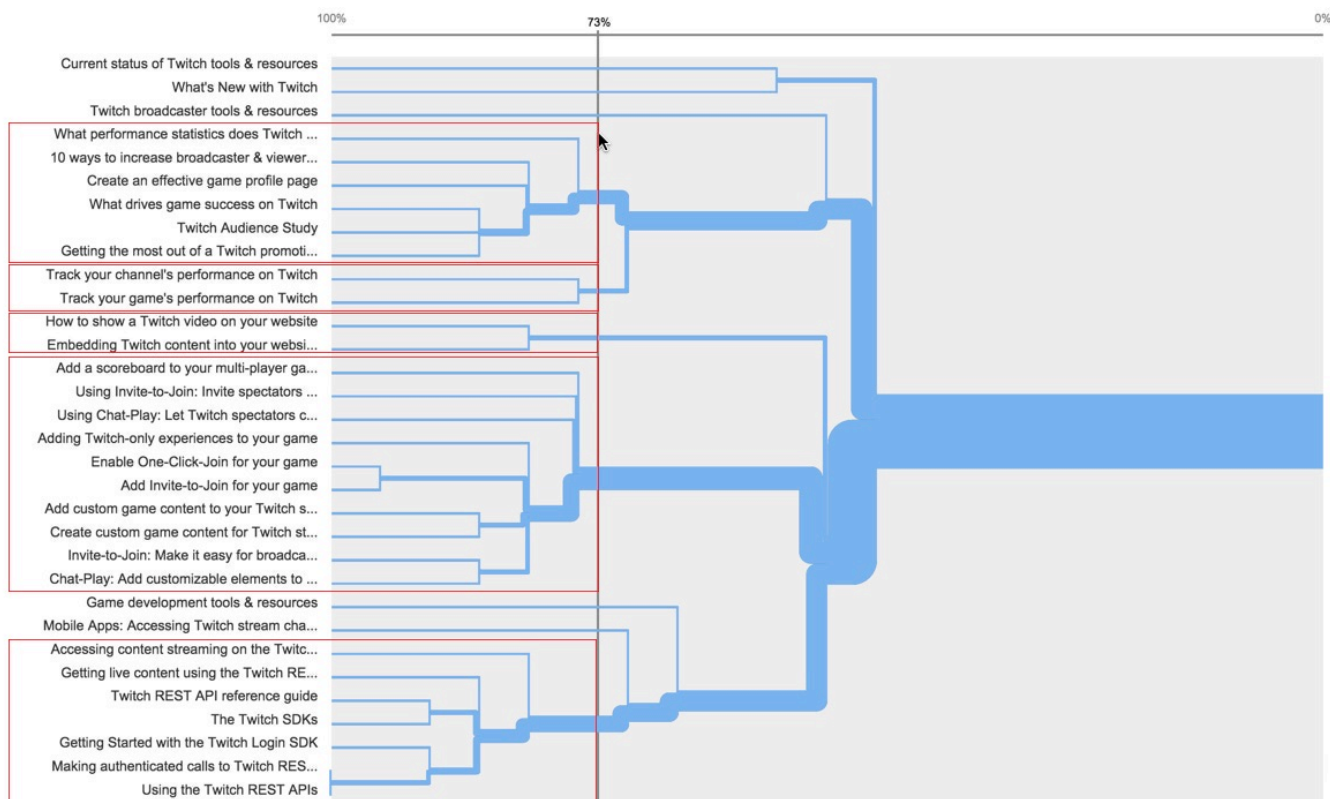
Topics each game-making role found relevant						
Topics	Producers	Junior Developers	Senior Game Developers	Designers	Broadcasters	Marketers
What's new with Twitch	✓			✓	✓	
Current status of Twitch tools & resources	✓	✓				
Twitch broadcaster tools & resources	✓	✓		✓	✓	
Game development tools & resources	✓	✓		✓		

The Twitch SDKs		✓				
Twitch REST API reference guide		✓	✓			
Getting Started with the Twitch Login SDK		✓	✓			
Using the Twitch REST APIs		✓	✓			
Making authenticated calls to Twitch REST APIs		✓	✓			
Twitch Audience Study	✓			✓	✓	✓
What drives game success on Twitch	✓	✓		✓	✓	✓
10 ways to increase broadcaster & viewer engagement with your game	✓	✓		✓		✓
Create an effective game profile page		✓		✓		✓
Adding Twitch-only experiences to your game	✓	✓		✓		✓
Create custom game content for Twitch streams	✓	✓		✓		✓
Add custom game content to your Twitch streams	✓	✓		✓	✓	✓
Getting the most out of a Twitch promotional event	✓	✓		✓		
Add a scoreboard to your multi-player game streams	✓	✓		✓		

Using Chat-Play: Let Twitch spectators customize your broadcast gameplay through chat	✓	✓		✓		✓
Chat-Play: Add customizable elements to your game for Twitch spectators to control through chat	✓	✓		✓		✓
Invite-to-Join: Make it easy for broadcasters to invite spectators to instantly join their game session	✓	✓		✓		
Add Invite-to-Join for your game	✓	✓		✓		✓
Enable One-Click-Join for your game	✓	✓		✓		✓
Using Invite-to-Join: Invite spectators to join your game session		✓		✓		
What performance statistics does Twitch provide and what do they mean?	✓			✓	✓	✓
Track your game's performance on Twitch	✓	✓		✓		✓
Track your channel's performance on Twitch	✓	✓		✓	✓	✓
Embedding Twitch content into your website, game or app	✓	✓			✓	
How to show a Twitch video on your website				✓		

Accessing content streaming on the Twitch IRC server		✓	✓			
Mobile Apps: Accessing Twitch stream chat & login by deeplinking to the Twitch mobile app						✓
Getting live content using the Twitch REST APIs						

## Card Sort Categorization Data



## User-Generated Topic Categories

1. Boost Your Presence on Twitch
2. Customize Your Game for Twitch
3. Track Performance on Twitch
4. Leverage Twitch to Promote Your Game
5. Engage with the Twitch API & SDK

## Proposed Topic Categories with Topics

Based on the user-generated categories, we determined appropriate names and the data categorization structure. We shifted around some of the topics that participants grouped together for clarity and organization. Below are the proposed topic categories.

### Develop Your Twitch Presence

- Understand your audience: Twitch Audience Study
- Cross-post: Show Twitch streams & videos on your website
- What drives game success on Twitch
- 10 ways to increase broadcaster & viewer engagement with your game
- Create an effective game profile page
- Getting the most out of a Twitch promotional event
- Create a game design/game dev stream
- Create an effective broadcaster profile page
- 10 ways to increase spectator engagement with your streams
- Take advantage of Twitch-specific game elements
  - Show custom game content in your Twitch streams
  - Add a scoreboard to your multi-player game streams
  - Using Invite-to-Join: Invite spectators to join your game session
  - Using Chat-Play: Let Twitch spectators customize your broadcast gameplay through chat

### Customize Your Game for Twitch

- Adding Twitch-only experiences to your game design
  - Invite-to-Join: Make it easy for broadcasters to invite spectators to instantly join their game session
  - Chat-Play: Add customizable elements to your game for Twitch spectators to control through chat
  - Create custom game content for Twitch streams
- Add Invite-to-Join for your game
- Enable One-Click-Join for your game
- Storing custom game content as broadcast data sources

### Add Twitch Content to Your App or Website

- Embedding Twitch content into your website, game or app
- Getting live content using the Twitch REST APIs
- Accessing content streaming on the Twitch IRC server
- Mobile Apps: Accessing Twitch stream chat & login by deeplinking to the Twitch mobile app -- Tools & Resources category?

### Track Performance on Twitch

- What performance statistics does Twitch provide and what do they mean?
- Track your game's performance on Twitch
- Track your channel's performance on Twitch
- Getting live performance stats from Twitch

### Tools & Resources

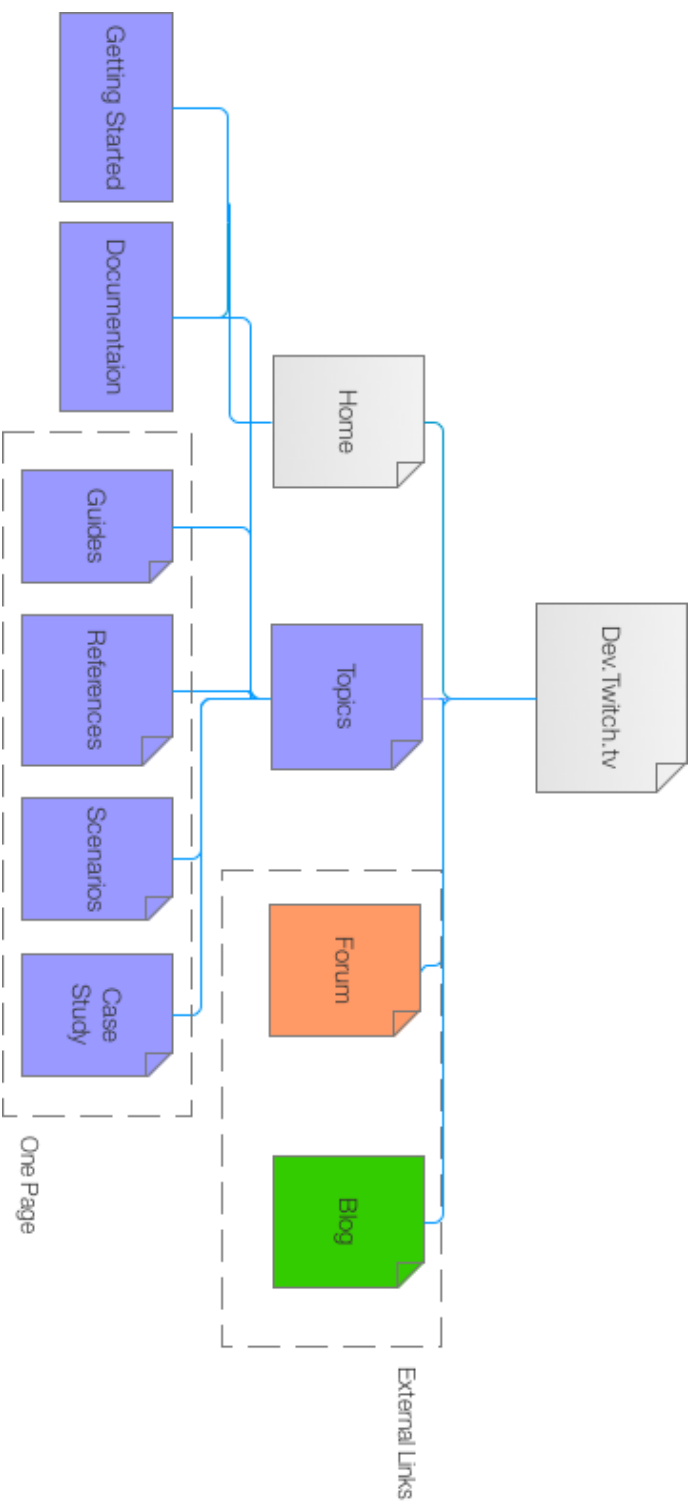
- Get Twitch tools & resources

- Twitch REST API
  - Overview
  - Using the Twitch REST APIs  
(includes general call how-to and making authenticated calls)
  - Twitch REST API reference
- Twitch Login SDK
  - Overview
  - Using the Twitch Login SDK
  - Twitch Login SDK reference
- Broadcaster tool/resource placeholder
  - Overview
  - Using
  - Reference



# **APPENDIX E: UX DESIGN SPEC & CUSTOMER SCENARIOS**

## **FOCUSED DESIGN**



## Get Started

Tools, Resources, and Ideas for Game Creators

Learn about the Twitch Audience

Creating Twitch Plugins & Apps

How to Get and Use Twitch Activity Stats

Game Makers: design Twitch-specific content

Using the Twitch REST APIs



Prominent Search

Call to Action

## Documentation

Create Awesome Experiences with Twitch

Browsing

Develop Your Twitch Presence

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum

Customize Your Game For Twitch

Add Twitch Content to Your App or Website

Track Performance on Twitch

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum

Tools and Resources

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum

Understand your audience: Twitch Audience Study

Cross-post: Show Twitch streams & videos on your website

What drives game success on Twitch

Read More ▼

Add invite-to-join for your game

Enable One-Click-join for your game

Read More ▼

Embedding Twitch content into your website, game or app

Getting live content using the Twitch REST APIs

Accessing content streaming on the Twitch IRC server

Read More ▼

Track your game's performance on Twitch

Track your channel's performance on Twitch

Read More ▼

Twitch REST API

Twitch Login SDK

Read More ▼

All Categories listed, links to Article Landing Page

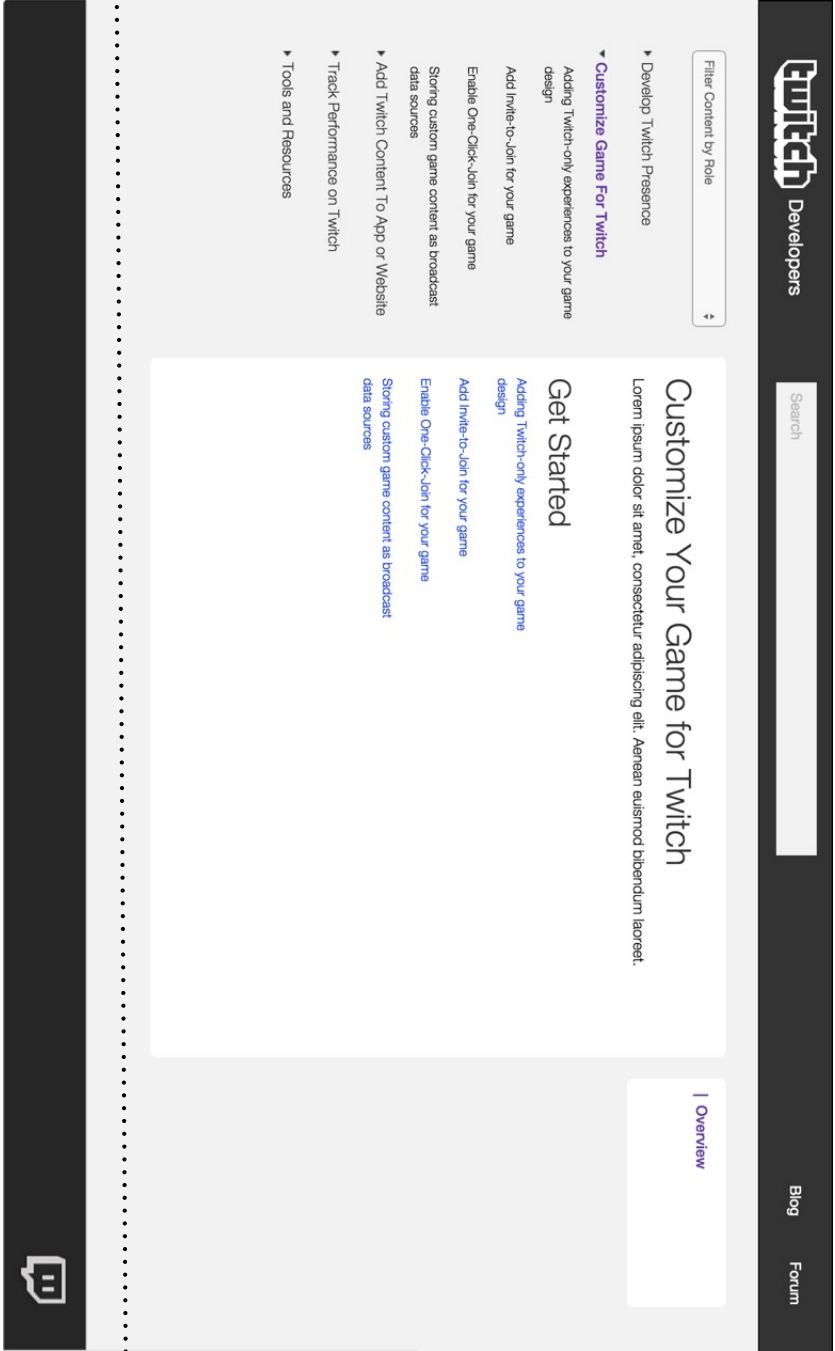
Links to Article

Links to Article Landing Page

Back to Twitch.tv



Article Landing Page



Visually group  
content

Task-driven  
information  
architecture

Game-maker  
role filter  
system

Filter Content by Role

- Develop Twitch Presence
- Customize Game For Twitch
- Add Twitch Content To App or Website
- Track Performance on Twitch

**Tools and Resources**

- Get Twitch tools & resources
- [Twitch REST API](#)
- [Twitch Login SDK](#)
- [Broadcaster tool/resource placeholder](#)

## Twitch REST API

### Calling a Twitch REST API

This section provides key guidance and recommendations for calling any of the Twitch REST APIs.

- Set the `Twitch-specific MIME type`. Set the `MIME type` using the `Accept` HTTP header as shown below. In response, the returned `MIME type` will always be `application/json`.

```
Accept: application/vnd.twitchtv.v3+json
```

- Identify the API version. If you don't specify a version, the latest stable version will be used. We recommend always specifying a version to ensure that version updates won't break your application.

To specify a version for a request, append the API version you want to use to the `Accept` header using the following format: `application/vnd.twitchtv(version)+json`. For example:

```
Accept: application/vnd.twitchtv.v3+json
```

If you can't use headers, specify the API version using a query string parameter, as follows:

```
https://api.twitch.tv/kraken/channels/hero?api_version=3
```

- Include your application's client ID in every request. This information is required to ensure that your application is not rate limited.

Set the client ID using the `Client-ID` HTTP header as follows:

```
Client-ID: <client_id>
```

If you can't use headers, specify the client ID using a query string parameter, as follows:

```
https://api.twitch.tv/kraken/channels/hero?client_id=<client_id>
```

- Enable authentication. Many of the Twitch APIs require that the caller be authorized to make the request, including those that access user account information or submit changes on a user's behalf. Authentication is handled by getting an access token and including it in each API request. See [Setting up Authentication](#) for more information on acquiring access tokens.

Once you have an access token, you can include it in your request in any of three ways.

- Include the token in the request header, using the `Authorization` HTTP header:

```
Authorization: OAuth [access token]
```

- Include the token in the request header, using the `Authorization` HTTP header:

```
auth_token=[access token]
```

- Send the token as a URL parameter:

```
https://api.twitch.tv/kraken/oauth_token?access_token=[access token]
```

**Overview**

- [Calling a Twitch REST API](#)
- [Getting an Access Token](#)

**Related Content**

- Article | [Twitch Login SDK](#)
- Article | [Getting Started with the Twitch Login SDK](#)
- Article | [Getting Started with the Twitch REST API](#)
- Discussion | [Bespoke Unity Integration Using the REST API](#)
- Discussion | [Hi, I've got a problem with the Twitch REST API](#)

Have Another Question?

[Ask Twitch Forum](#)

**Previously Viewed**

- [Using the Twitch REST APIs](#)
- [Making authenticated calls to Twitch REST API](#)

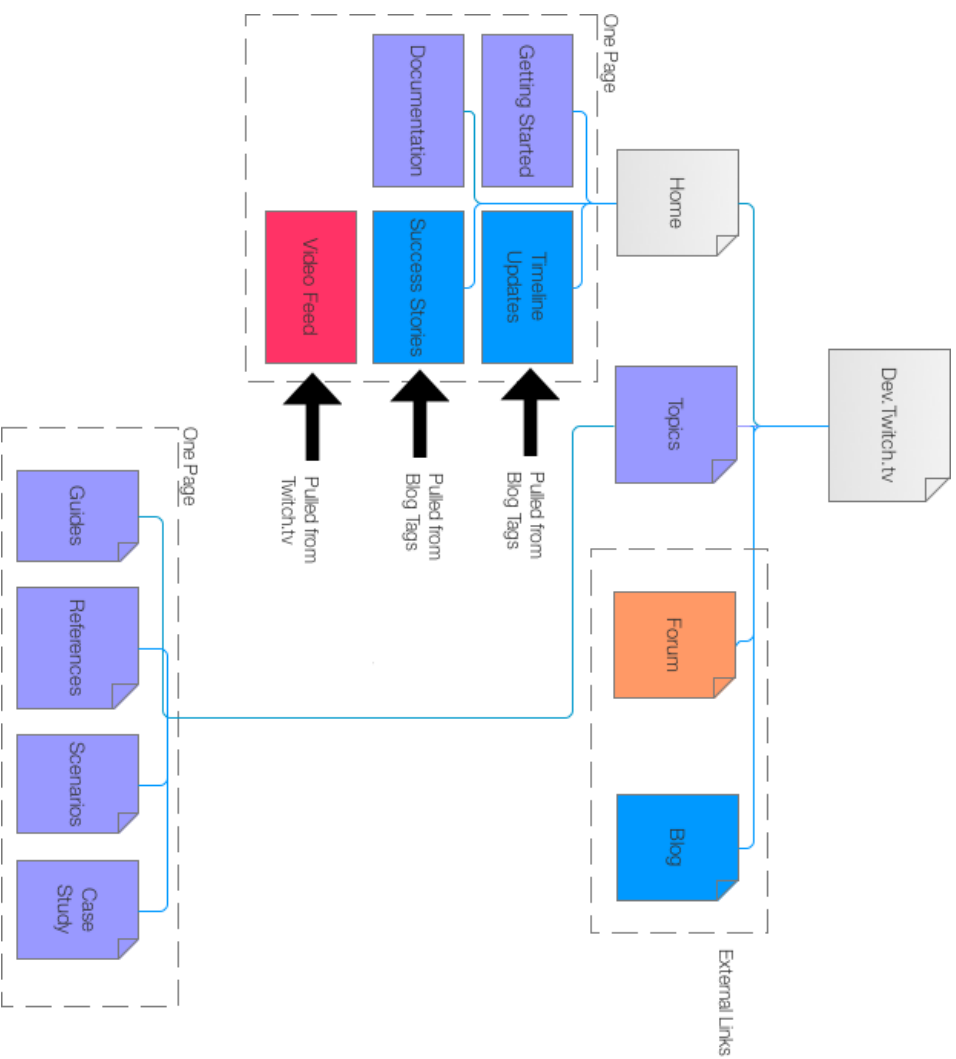
Table of contents for page

Articles + Forum links  
Content pulled by tagging system

Code samples  
Easy to copy & paste

Page feedback

## **SCALABLE DESIGN**





## Tools, Resources, and Ideas for Game Creators

**Choice Chamber: The First Game Built For Twitch**  
August 16, 2015, 3:05 pm



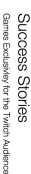
Create Awesome Experiences with Twitter

Create Awesome Experiences with Twitter

- ▶ **Develop Your Twitch Presence**  
Learn how to set up your Twitch channel, connect your Twitch account to your YouTube channel, and more.
- ▶ **Customize Your Game For Twitch**  
Learn how to set up your game for Twitch, connect your Twitch account to your game, and more.
- ▶ **Add Twitch Content to Your App or Website**  
Learn how to add Twitch content to your app or website, connect your Twitch account to your app or website, and more.
- ▶ **Track Performance on Twitch**  
Learn how to track your Twitch performance, connect your Twitch account to your analytics, and more.

## Twitch Game Development Office Hours

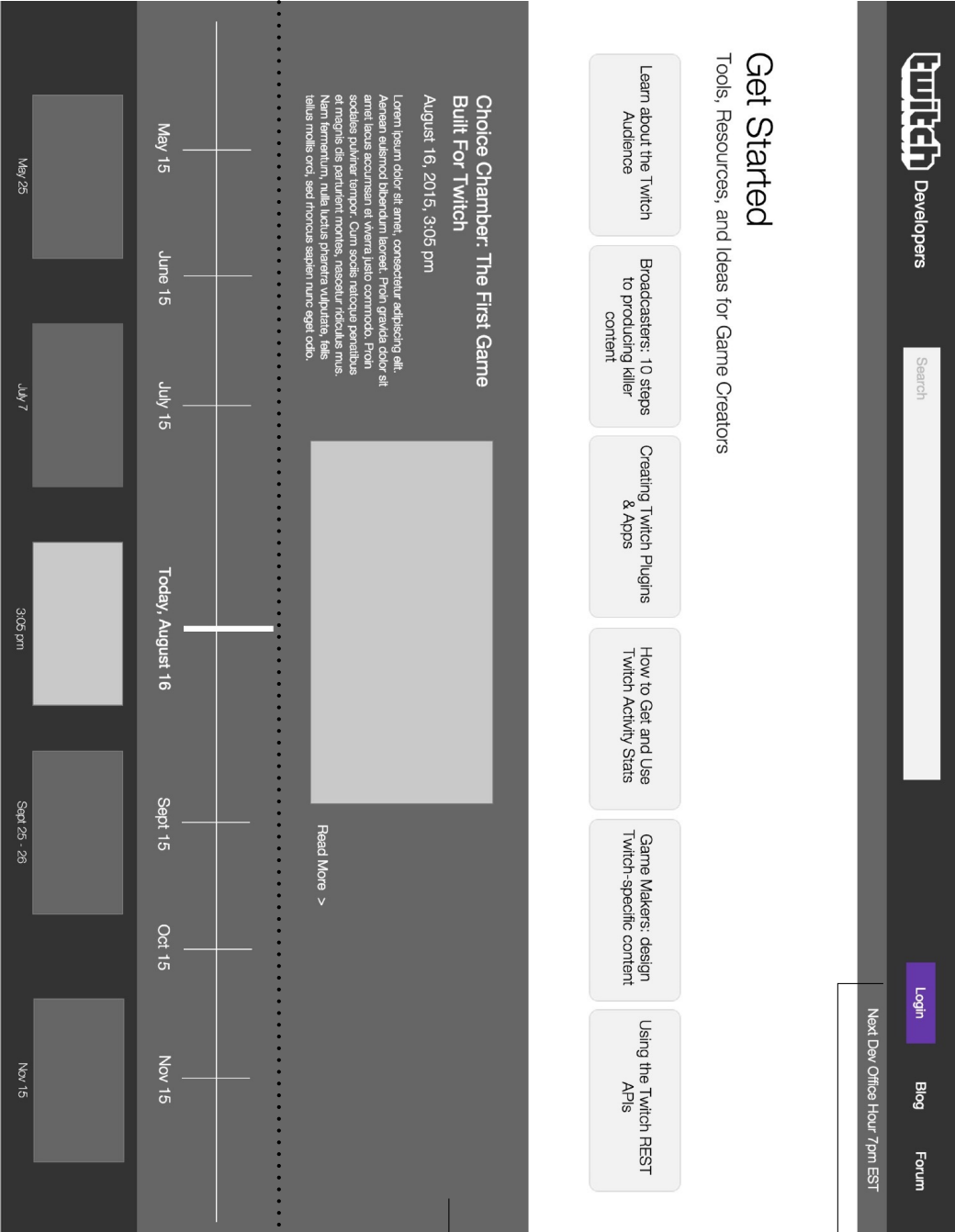
## Twitch Game Development Office Hours



## Games Exclusively for the Twitch Audience

## Games Exclusively for the Twitch Audience





Twitch Identity for video feed

Call to Action

Timeline of Product & Features

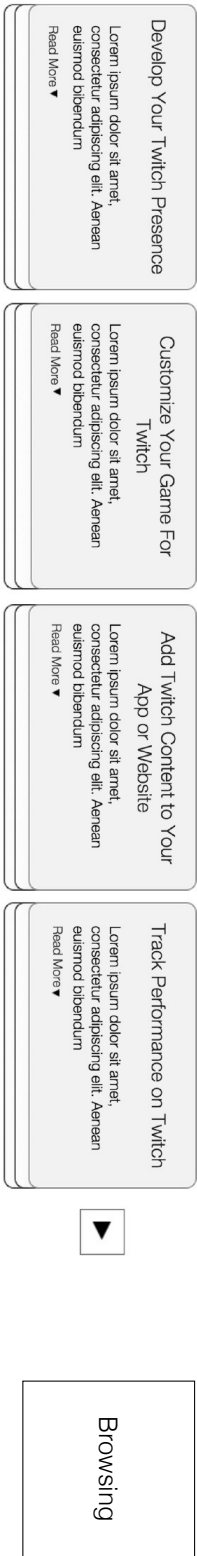
Links to blog

Toggle between posts by moving date cursor or clicking on image below timeline

Content pulled from  
blog.twitch.tv/category/dev

# Documentation

Create Awesome Experiences with Twitch



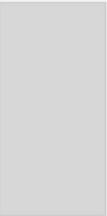


Live feed of from Game Development Feed  
Twitch hosts live office hours

Three additional video feeds

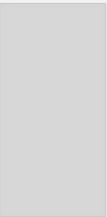
# Success Stories

Games Exclusively for the Twitch Audience



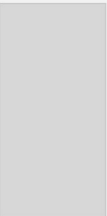
ESL has made a one stop shop for eSports content by making great use of Twitch streams, videos, and metadata. From building informative video overlays, to bringing together event scheduling with content, to integrating it all in one package, ESL keeps their audience captivated.

[Read More >](#)



Bot used to moderate Twitch chats, an indispensable tool for broadcasters who need to built the strength of their moderation team. Moobot processes 1+ million Twitch chat messages a day for over 5,000 broadcasters.

[Read More >](#)



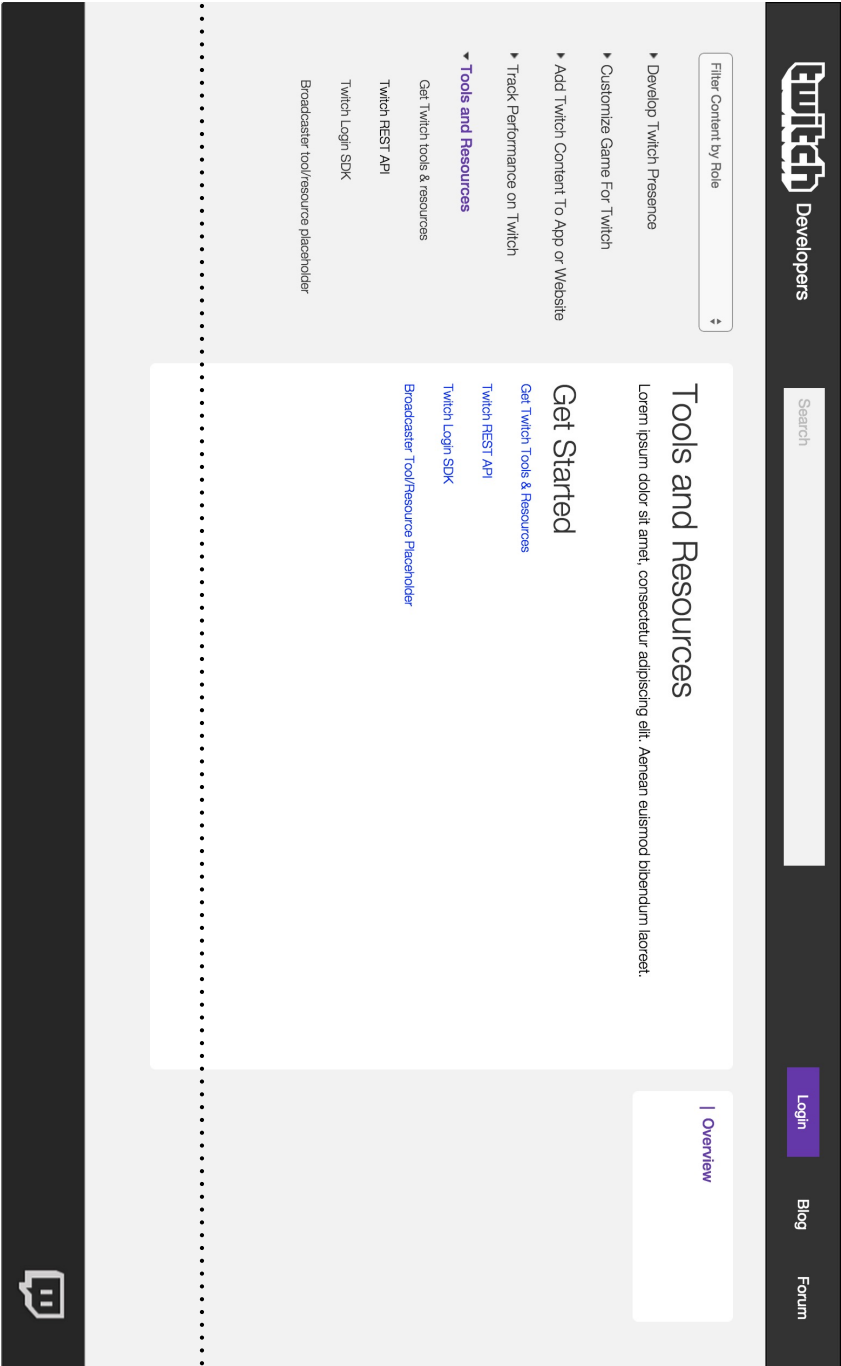
Twitch Now is a much loved and 5 star rated Chrome browser extension. Built entirely upon our web API, Twitch Now keeps users updated and connected to the Twitch experience outside of our website.

[Read More >](#)

[Links to Blog](#)



[Back to Twitch.tv](#)



Article Landing Page

Filter Content by Role

- Develop Twitch Presence
- Customize Game For Twitch
- Add Twitch Content To App or Website
- Track Performances on Twitch

#### Tools and Resources

Get Twitch tools & resources

[Twitch REST API](#)

[Twitch Login SDK](#)

[Broadcaster tool/resource placeholder](#)

## Twitch REST API

### Calling a Twitch REST API

This section provides key guidance and recommendations for calling any of the Twitch REST APIs.

1. Set the `Twitch` specific `NAME` type. Set the `NAME` type using the `Accept` HTTP header as shown below. In response, the returned `NAME` type will always be `application/json`.

```
Accept: application/vnd.twitchtv.v3+json
```

2. Identify the API version. If you don't specify a version, the latest stable version will be used. We recommend always specifying a version to ensure that version updates won't break your application.

To specify a version for a request, append the API version you want to use to the `Accept` header using the following format: `application/vnd.twitchtv.version+json`. For example:

```
Accept: application/vnd.twitchtv.v3+json
```

If you can't use headers, specify the API version using a query string parameter, as follows:

```
https://api.twitch.tv/kraken/channels/nbro?api_version=3
```

3. Include your application's client ID in every request. This information is required to ensure that your application is not rate limited.

Set the client ID using the `Client-ID` HTTP header as follows:

```
Client-ID: <client_id>
```

If you can't use headers, specify the client ID using a query string parameter, as follows:

```
https://api.twitch.tv/kraken/channels/nbro?client_id=<client_id>
```

4. Enable authentication. Many of the Twitch APIs require that the caller be authorized to make the request, including those that access user account information or submit changes on a user's behalf. Authentication is handled by getting an access token and including it in each API request. See "Setting Up Authentication" for more information on acquiring access tokens.

Once you have an access token, you can include it in your request in any of three ways:

- Include the token in the request header, using the `Authorization` HTTP header:  
`Authorization: OAuth [access token]`

- Include the token in the request header, using the `Authorization` HTTP header:  
`auth_token=[access token]`

- Send the token as a URL parameter:  
`https://api.twitch.tv/kraken/oauth_token?access_token=[access token]`

[Overview](#)  
| [Calling a Twitch REST API](#)  
| [Getting an Access Token](#)

#### Related Content

Article | [Twitch Login SDK](#)

Article | [Getting Started with the Twitch Login SDK](#)

Article | [Getting Started with the Twitch REST API](#)

Discussion | [Bespoke Unity Integration Using the REST API](#)

Discussion | ["Hi, I've got a problem with the Twitch REST API"](#)

[Have Another Question?](#)

[Ask Twitch Forum](#)

#### Previously Viewed

[Using the Twitch REST APIs](#)

[Making authenticated calls to Twitch REST API](#)

Table of contents for page

Articles + Forum links

Content pulled by tagging system

Code samples

Easy to copy & paste

Page feedback

Have a Question or Comment?

[Give Feedback](#)



## **CUSTOMER SCENARIO: GAME DESIGNER LOOKS FOR IDEAS**





Kim wants to optimize her game for Twitch but doesn't know where to start

Kim arrives on the home page from a Google Search. She scans the content & clicks on Getting Started

Getting Started takes Kim to an article page

Kim scans the page and looks at other articles in the category

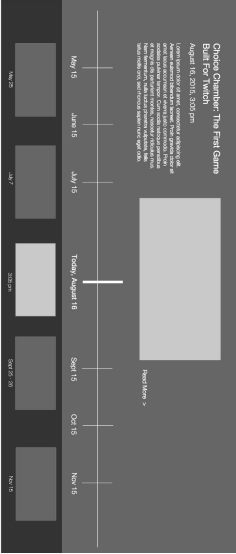
She finds a cool idea and shares the page link with her colleagues



# Get Started

Tools, Resources, and Ideas for Game Creators

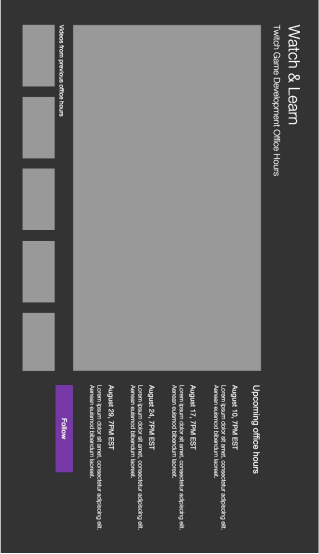
- Learn about the Twitch API
- Discover the Twitch API
- Check out the Twitch API
- Learn about the Twitch API
- Check out the Twitch API
- Learn about the Twitch API



# Documentation

Create Amazing Experiences with Twitch

- Getting Started with Twitch
- Getting Started with Twitch
- Getting Started with Twitch
- Getting Started with Twitch
- Getting Started with Twitch
- Getting Started with Twitch



# Success Stories

Games Exclusively for the Twitch Audience



## Get Started

Tools, Resources, and Ideas for Game Creators

Learn about the Twitch Audience

Broadcasters: 10 steps to producing killer content

Creating Twitch Plugins & Apps

How to Get and Use Twitch Activity Stats

Game Makers: design Twitch-specific content

Using the Twitch REST APIs

Choice Chamber: The First Game Built For Twitch

August 16, 2015, 3:05 pm

Filter Content by Role

► Develop Twitch Presence

▼ **Customize Game For Twitch**

Adding Twitch-only experiences to your game design

Add Invite-to-Join for your game

Enable One-Click-Join for your game

Storing custom game content as broadcast data sources

► Add Twitch Content To App or Website

► Track Performance on Twitch

► Tools and Resources

## Customize Your Game for Twitch

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet.

### Get Started

Adding Twitch-only experiences to your game design

Add Invite-to-Join for your game

Enable One-Click-Join for your game

Storing custom game content as broadcast data sources

| Overview



Filter Content by Role

► Develop Twitch Presence

► **Customize Game For Twitch**  
Adding Twitch-only experiences to your game design

Add links-to-join for your game

Enable One-Click-Join for your game

Storing custom game content as broadcast data sources

► Add Twitch Content To App or Website

► Track Performance on Twitch

► Tools and Resources

## Adding Twitch Only Experiences to Your Game

### Customize

1. Set the Twitch-specific MIME type. Set the MIME type using the Accept HTTP header as shown below. In response, the returned MIME type will always be application/json.

2. Identify the API version. If you don't specify a version, the latest stable version will be used. We recommend always specifying a version to ensure that version updates won't break your application.

To specify a version for a request, append the API version you want to use to the Accept header using the following format: application/vnd.twitchtv+version+json. For example:

If you can't use headers, specify the API version using a query string parameter, as follows:

Include your application's client ID in every request. This information is required to ensure that your application is not rate limited.

Set the client ID using the Client-ID HTTP header as follows:

3. If you can't use headers, specify the client ID using a query string parameter, as follows:

4. Enable authentication. Many of the Twitch APIs require that the caller be authorized to make the request, including those that access user account information or submit charges on a user's behalf. Authentication is handled by getting an access token and including it in each API request. See "Setting up Authentication" for more information on acquiring access tokens.

Once you have an access token, you can include it in your request in any of three ways:

- Include the token in the request header, using the Authorization HTTP header.
- Include the token in the request header, using the Authorization HTTP header.
- Send the token as a URL parameter.

| Overview  
| Customize  
| Adding Features

#### Related Content

Article | Twitch Login SDK

Article | Getting Started with the Twitch Login SDK

Article | Getting Started with the Twitch REST API

Discussion | Bespoke Unity Integration Using the REST API

Discussion | Hi, I've got a problem with the Twitch REST API

Have Another Question?

Ask Twitch Forum

#### Previously Viewed

Using the Twitch REST APIs

Making authenticated calls to Twitch REST API

Have a Question or Comment?

Give Feedback



Filter Content by Role



▶ Develop Twitch Presence

▶ **Customize Game For Twitch**

*Adding Twitch-only experiences to your game design*

Add Invite-to-Join for your game

Enable One-Click-Join for your game

Storing custom game content as broadcast data sources

▶ Add Twitch Content To App or Website

▶ Track Performance on Twitch

▶ Tools and Resources

## Adding Twitch Only Experiences to Your Game

### Customize

1. Set the Twitch-specific MIME type. Set the MIME type using the Accept HTTP header as shown below. In response, the returned MIME type will always be application/json.

2. Identify the API version. If you don't specify a version, the latest stable version will be used. We recommend always specifying a version to ensure that version updates won't break your application.

To specify a version for a request, append the API version you want to use to the Accept header using the following format: application/vnd.twitch.v{version}-json. For example:

If you can't use headers, specify the API version using a query string parameter, as follows:

Include your application's client ID in every request. This information is required to ensure that your application is not rate limited.

Set the client ID using the Client-ID HTTP header as follows:

3. If you can't use headers, specify the client ID using a query string parameter, as follows:

4. Enable authentication. Many of the Twitch APIs require that the caller be authorized to make the request, including those that access user account information or submit changes on a user's behalf. Authentication is handled by getting an access token and including it in each API request. See [Setting up Authentication](#) for more information on acquiring access tokens.

Once you have an access token, you can include it in your request in any of three ways:

- Include the token in the request header, using the Authorization HTTP header.
- Include the token in the request header, using the Authorization HTTP header.
- Send the token as a URL parameter.

- | Overview
- | Customize
- | Adding Features

#### Related Content

Article | Twitch Login SDK

Article | Getting Started with the Twitch Login SDK

Article | Getting Started with the Twitch REST API

Discussion | Bepoche Unity Integration Using the REST API

Discussion | Hi, I've got a problem with the Twitch REST API

Have Another Question?

Ask Twitch Forum

#### Previously Viewed

Using the Twitch REST APIs

Making authenticated calls to Twitch REST API

Have a Question or Comment?

Give Feedback



Filter Content by Role

► Develop Twitch Presence

► **Customize Game For Twitch**

Adding Twitch-only experiences to your game design

[Add Invite-to-Join for your game](#)

Enable One-Click-Join for your game

Storing custom game content as broadcast data sources

► Add Twitch Content To App or Website

► Track Performance on Twitch

► Tools and Resources

## Add Invite-to-Join for your game

### Enable Invite

1. Set the Twitch-specific MIME type. Set the MIME type using the Accept HTTP header as shown below. In response, the returned MIME type will always be application/json.
2. Identify the API version. If you don't specify a version, the latest stable version will be used. We recommend always specifying a version to ensure that version updates won't break your application.

To specify a version for a request, append the API version you want to use to the Accept header using the following format: application/vnd.twitchtv+json. For example:

If you can't use headers, specify the API version using a query string parameter, as follows:

Include your application's client ID in every request. This information is required to ensure that your application is not rate limited.

Set the client ID using the Client-ID HTTP header as follows:

3. If you can't use headers, specify the client ID using a query string parameter, as follows:

4. Enable authentication. Many of the Twitch APIs require that the caller be authorized to make the request, including those that access user account information or submit changes on a user's behalf. Authentication is handled by getting an access token and including it in each API request. See "Setting up Authentication" for more information on acquiring access tokens.

Once you have an access token, you can include it in your request in any of three ways:

- Include the token in the request header, using the Authorization HTTP header.
- Include the token in the request header, using the Authorization HTTP header.
- Send the token as a URL parameter.

- | Overview
- | Enable Invite
- | Adding Features

#### Related Content

Article | Twitch Login SDK

Article | Getting Started with the Twitch Login SDK

Article | Getting Started with the Twitch REST API

Discussion | Bespoke Unity Integration Using the REST API

Discussion | Hi, I've got a problem with the Twitch REST API

Have Another Question?

[Ask Twitch Forum](#)

#### Previously Viewed

Using the Twitch REST APIs

Making authenticated calls to Twitch REST API

Have a Question or Comment?

[Give Feedback](#)



## **CUSTOMER SCENARIO: GAME DEVELOPER SEARCH FOR A QUICK ANSWER**





Carlos needs to find the Twitch REST API reference list

He types some key words into a Google Search

The first hit takes him to a landing page in Twitch documentation

Carlos uses a quick link on the page to navigate to the API information

After the page loads Carlos searches for a keyword using control F

Carlos doesn't see the object he needs. A discussion link catches his eye & he clicks on it



Filter Content by Role



► Develop Twitch Presence

► Customize Game For Twitch

► Add Twitch Content To App or Website

► Track Performance on Twitch

▼ **Tools and Resources**

Get Twitch tools & resources

Twitch REST API

Twitch Login SDK

Broadcaster tool/resource placeholder

## Tools and Resources

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet.

### Get Started

[Get Twitch Tools & Resources](#)

[Twitch REST API](#)

[Twitch Login SDK](#)

[Broadcaster Tool/Resource Placeholder](#)

| [Overview](#)



Filter Content by Role

Develop Twitch Presence

Customize Game For Twitch

Add Twitch Content To App or Website

Track Performance on Twitch

Tools and Resources

Get Twitch tools & resources

Twitch REST API

Twitch Login SDK

Broadcaster tool/resource placeholder

## Twitch REST API

### Calling a Twitch REST API

This section provides key guidance and recommendations for calling any of the Twitch REST APIs.

1. Set the `Twitch-specific MIME type`. Set the `MIME type` using the `Accept` HTTP header as shown below. In response, the returned `MIME type` will always be `application/json`.

```
Accept: application/vnd.twitchtv.v3+json
```

2. Identify the API version. If you don't specify a version, the latest stable version will be used. We recommend always specifying a version to ensure that version updates won't break your application.

To specify a version for a request, append the API version you want to use to the `Accept` header using the following format: `application/vnd.twitchtv.version+json`. For example:

```
Accept: application/vnd.twitchtv.v3+json
```

If you can't use headers, specify the API version using a query string parameter, as follows:

```
https://api.twitch.tv/kraken/channels/hbo?client_id=client_id&version=3
```

3. Include your application's client ID in every request. This information is required to ensure that your application is not rate limited.

Set the client ID using the `Client-ID` HTTP header as follows:

```
Client-ID: <client_id>
```

If you can't use headers, specify the client ID using a query string parameter, as follows:

```
https://api.twitch.tv/kraken/channels/hbo?client_id=client_id
```

4. Enable authentication. Many of the Twitch APIs require that the caller be authorized to make the request, including those that access user account information or submit changes to a user's profile. Authentication is handled by getting an access token and including it in each API request. See [Setting up Authentication](#) for more information on acquiring access tokens.

Once you have an access token, you can include it in your request in any of three ways:

• Include the token in the request header, using the `Authorization` HTTP header:

```
Authorization: OAuth [access_token]
```

• Include the token in the request header, using the `Authorization` HTTP header:

```
auth_token= [access_token]
```

• Send the token as a URL parameter:

```
https://api.twitch.tv/kraken/oauth_token?access_token= [access_token]
```

[Overview](#)  
| [Calling a Twitch REST API](#)  
| [Getting an Access Token](#)

#### Related Content

Article | [Twitch Login SDK](#)

Article | [Getting Started with the Twitch Login SDK](#)

Article | [Getting Started with the Twitch REST API](#)

Discussion | [Bespoke Unity Integration Using the REST API](#)

Discussion | [Hi, I've got a problem with the Twitch REST API](#)

[Have Another Question?](#)

[Ask Twitch Forum](#)

#### Previously Viewed

[Using the Twitch REST APIs](#)

[Making authenticated calls to Twitch REST API](#)

[Have a Question or Comment?](#)

[Give Feedback](#)



Filter Content by Role

- Develop Twitch Presence
- Customize Game For Twitch
- Add Twitch Content To App or Website
- Track Performances on Twitch

## Tools and Resources

- Get Twitch tools & resources
- [Twitch REST API](#)
- [Twitch Login SDK](#)
- [Broadcaster tool/resources placeholder](#)

## Twitch REST API

### Calling a Twitch REST API

This section provides key guidance and recommendations for calling any of the Twitch REST APIs.

1. Set the `Twitch` specific `NAME` type. Set the `NAME` type using the `Accept` HTTP header as shown below. In response, the returned `NAME` type will always be `application/json`.

```
Accept: application/vnd.twitchtv.v3+json
```

2. Identify the API version. If you don't specify a version, the latest stable version will be used. We recommend always specifying a version to ensure that version updates won't break your application.

To specify a version for a request, append the API version you want to use to the `Accept` header using the following format: `application/vnd.twitchtv.version+json`. For example:

```
Accept: application/vnd.twitchtv.v3+json
```

If you can't use headers, specify the API version using a query string parameter, as follows:

```
https://api.twitch.tv/kraken/channels/nb0?client_id=<client_id>&version=3
```

3. Include your application's client ID in every request. This information is required to ensure that your application is not rate limited.

Set the client ID using the `Client-ID` HTTP header as follows:

```
Client-ID: <client_id>
```

If you can't use headers, specify the client ID using a query string parameter, as follows:

```
https://api.twitch.tv/kraken/channels/nb0?client_id=<client_id>
```

4. Enable authentication. Many of the Twitch APIs require that the caller be authorized to make the request, including those that access user account information or submit changes on a user's behalf. Authentication is handled by getting an access token and including it in each API request. See [Setting Up Authentication](#) for more information on acquiring access tokens.

Once you have an access token, you can include it in your request in any of three ways.

- Include the token in the request header, using the `Authorization` HTTP header.  
`Authorization: OAuth [access token]`

- Include the token in the request header, using the `Authorization` HTTP header.  
`auth_token=[access token]`

- Send the token as a URL parameter.  
`https://api.twitch.tv/kraken/oauth_token?access_token=[access token]`

- | [Overview](#)
- | [Calling a Twitch REST API](#)
- | [Getting an Access Token](#)

### Related Content

- Article | [Twitch Login SDK](#)
- Article | [Getting Started with the Twitch Login SDK](#)
- Article | [Getting Started with the Twitch REST API](#)
- Discussion | [Bespoke Unity Integration Using the REST API](#)

Discussion | ["Hi, I've got a problem with the Twitch REST API"](#)

Have Another Question?

[Ask Twitch Forum](#)

### Previously Viewed

- [Using the Twitch REST APIs](#)
- [Making authenticated calls to Twitch REST API](#)

discuss.dev.twitch.tv



## **CUSTOMER SCENARIO: GAME PRODUCER FINDS PRODUCT ROADMAP**



Mark is planning his development roadmap and needs to know when Twitch will release new features

Mark arrives on the home page from a Google Search

The timeline immediately catches his eye

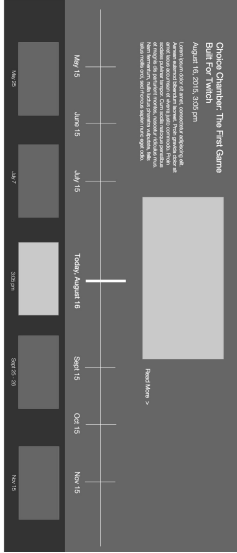
Mark plays around with the timeline, reading the recent updates & older updates

Mark looks at the prospective timeline dates & finds relevant information

## Get Started

Tools, Resources, and Ideas for Game Creators

- [Learn about the Twitch Audience](#)
- [Broadcasting: 10 Steps to getting started](#)
- [Creating Twitch Plugins & Apps](#)
- [How to Get and Use Twitch Activity Stats](#)
- [Game Manager: Design Twitch-specific content](#)
- [Using the Twitch REST API](#)



## Documentation

Create Amazing Experiences with Twitch

- [Design Your Twitch Presence](#)  
Learn how to design your Twitch presence, from your channel art to your stream overlay, to create a cohesive and engaging experience for your audience.
- [Customize Your Game for Twitch](#)  
Learn how to customize your game for Twitch, from adding Twitch-specific content to your game to creating a Twitch-specific experience for your audience.
- [Add Twitch Content to Your App or Website](#)  
Learn how to add Twitch content to your app or website, from adding Twitch-specific content to your app or website to creating a Twitch-specific experience for your audience.
- [Track Performance on Twitch](#)  
Learn how to track performance on Twitch, from adding Twitch-specific content to your app or website to creating a Twitch-specific experience for your audience.



## Success Stories

Games Exclusively for the Twitch Audience

- [The Game of the Year: A Game Exclusively for the Twitch Audience](#)  
Learn how to create a game exclusively for the Twitch audience, from the design to the development, to the marketing and distribution.
- [The Game of the Year: A Game Exclusively for the Twitch Audience](#)  
Learn how to create a game exclusively for the Twitch audience, from the design to the development, to the marketing and distribution.
- [The Game of the Year: A Game Exclusively for the Twitch Audience](#)  
Learn how to create a game exclusively for the Twitch audience, from the design to the development, to the marketing and distribution.



## Get Started

Tools, Resources, and Ideas for Game Creators

Learn about the Twitch Audience

Broadcasters: 10 steps to producing killer content

Creating Twitch Plugins & Apps

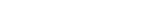
How to Get and Use Twitch Activity Stats

Game Makers: design Twitch-specific content

Using the Twitch REST APIs

Choice Chamber: The First Game Built For Twitch

August 16, 2015, 3:05 pm



Choice Chamber: The First Game  
Built For Twitch

August 16, 2015, 3:05 pm

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo. Proin sodales pulvinar tempor. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nam fermentum, nulla luctus pharetra vulputate, felis tellus mollis orci, sed rhoncus sapien nunc eget odio.

Read More >

May 15

June 15

July 15

Today, August 16

Sept 15

Oct 15

Nov 15

May 25

July 7

3:05 pm

Sept 25 - 28

Nov 15

Goes to Blog

Returns

## TwitchCon 2015 Schedule

July 9, 2015, 11:40 am

Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
 Aenean euismod bibendum laoreet. Proin gravida dolor sit  
 amet lacus accumsan et viverra nulla commodo. Proin  
 sodales pulvinar tempor. Cum sociis natoque penatibus  
 et magnis dis parturient montes, nascetur ridiculus mus.  
 Nam fermentum, nulla luctus pharetra vulputate, felis  
 tellus mollis orci, sed rhoncus sapien nunc eget odio.

[Read More >](#)

May 15

June 15

July 15

Today, August 16

Sept 15

Oct 15

Nov 15

May 25

July 7

3:05 pm

Sept 25 - 26

Nov 15

## Goes to Blog