

# Twitch Alert

Travail de semestre

---

CFPT Informatique  
Jean-Philippe Froelicher  
T.IS-E2A

2014

# Table des matières

<b>I</b>	<b>Cahier des charges</b>	<b>3</b>
<b>1</b>	<b>Initial</b>	<b>3</b>
1.1	Titre du projet . . . . .	3
1.2	Objectifs du projet . . . . .	3
1.3	Description du projet . . . . .	3
<b>II</b>	<b>Étude d'opportunité</b>	<b>5</b>
<b>2</b>	<b>Introduction au projet</b>	<b>5</b>
2.1	Twitch . . . . .	5
2.1.1	Chaîne . . . . .	5
2.1.2	Team . . . . .	5
2.1.3	Suivis (Following) . . . . .	5
2.2	Pourquoi avoir choisi ce sujet ? . . . . .	5
2.3	Pourquoi une application de bureau ? . . . . .	5
2.4	Pourquoi C# ? . . . . .	6
<b>3</b>	<b>Analyse de l'existant</b>	<b>6</b>
3.1	Existant . . . . .	6
3.2	Critique de l'existant . . . . .	6
<b>4</b>	<b>Analyse fonctionnelle</b>	<b>7</b>
4.1	Gestion des chaines . . . . .	7
4.2	Recherche d'une chaine . . . . .	7
4.3	Ajout d'une chaîne . . . . .	7
4.4	Gestion des notifications . . . . .	7
4.5	Interface homme-machine . . . . .	7
4.5.1	Vue principale . . . . .	7
<b>5</b>	<b>Analyse organique</b>	<b>10</b>
5.1	Diagramme de classe . . . . .	10
5.2	API Twitch (Twitch.Net) . . . . .	10
5.2.1	OAuth 2.0 . . . . .	10
5.2.2	Site d'autorisation . . . . .	11
5.2.3	cURL . . . . .	11
<b>6</b>	<b>Tests</b>	<b>12</b>
<b>7</b>	<b>Conclusion</b>	<b>12</b>
<b>8</b>	<b>Glossaire</b>	<b>12</b>
<b>9</b>	<b>Annexes</b>	<b>12</b>
9.1	Planification . . . . .	12
9.1.1	Planification prévisionnelle . . . . .	12
9.1.2	Planning réel . . . . .	13

## Première partie

# Cahier des charges

## 1 Initial

### 1.1 Titre du projet

*Twitch Alert*

### 1.2 Objectifs du projet

Créer une application de notification(alerte) pour la plateforme de diffusion de *streaming* **TWITCH**.

Apprentissage et prise en main de l'API <sup>1</sup> Twitch.

### 1.3 Description du projet

#### Glossaire :

stream (n.m) : Flux vidéo.

streamer (v.) : L'action de diffuser du flux vidéo.

streaming (n.m) : Diffusion de vidéos en continu.

streameur (n.m) : Personne diffusant une vidéo en continu.

Twitch : Site permettant à n'importe qui de diffuser du flux vidéo, chaque utilisateur a une chaine Twitch permettant de diffuser du flux vidéo.

Chaîne Twitch : Un utilisateur Twitch a forcément une chaine Twitch qu'il utilise que lorsqu'il diffuse du flux vidéo.

#### Général :

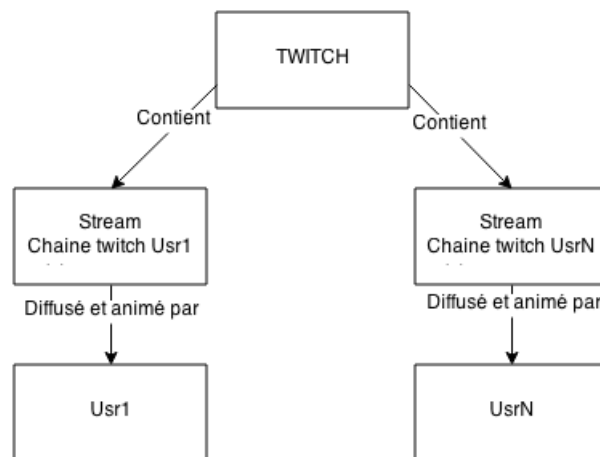
Une vue principale qui permet de configurer les différentes notifications et alertes pour chaque flux vidéo (stream).

L'application tourne en tâche de fond, c'est-à-dire qu'une icône apparait dans la barre de tâche lorsque l'on réduit la fenêtre de l'application. Pour y accéder on clique sur l'icône de la barre de tâche.

Une notification est représentée sous forme de *pop-up*(petite fenêtre) qui apparait en bas à droite de l'écran principal, un petit son est émis lorsque la notification apparait si l'utilisateur ne la voit pas.

---

1. Application Programming Interface



### Twitch

- La plateforme Twitch permet de suivre des streameurs "indépendants", actuellement, lorsqu'un streameur commence son stream, une notification est envoyée par e-mail. Le but est de pouvoir saisir le nom Twitch du streameur, et lorsque il commence son stream, mon application notifie l'utilisateur.
- Pouvoir regarder le stream d'un utilisateur Twitch.
- Gestion des notification pour chaque chaine Twitch.
- Gestion des plages horaires de notifications pour chaque chaine Twitch

## Deuxième partie

# Étude d'opportunité

## 2 Introduction au projet

Le but de ce projet est de réaliser une application de notification pour la plateforme de diffusion de streaming Twitch ainsi que toute la gestion de ses notifications.

### 2.1 Twitch

Twitch est une plateforme de diffusion de streaming et de VOD<sup>2</sup> de jeux vidéo créée en 2011. C'est la plateforme numéro un pour les amateurs de streaming, toutes personnes voulant commencer dans ce monde est quasiment obligé de passer par Twitch.

La plateforme donne la possibilité de pouvoir transmettre du flux vidéo sur la chaîne Twitch de l'utilisateur en question.

#### 2.1.1 Chaîne

Une chaîne Twitch offre une multitude de possibilité pour l'utilisateur spectateur, effectivement une chaîne n'est pas simplement un flux vidéo. Il y a la fonction "Suivre" une chaîne Twitch, qui permet aux spectateurs de pouvoir être tenu au courant via e-mail mais aussi d'accéder plus rapidement à la chaîne en question. Il y a également une fonction "S'abonner" qui est payante, le prix varie selon la chaîne. En revanche, cette fonction offre quelques avantages pas très significatifs car l'abonnement sert surtout à soutenir financièrement directement la chaîne Twitch.

#### 2.1.2 Team

Une team Twitch est un groupe de chaîne Twitch, la notion de team permet uniquement de retrouver les différentes chaîne d'un groupe de streaming, ou d'une team professionnelle de jeu etc...

#### 2.1.3 Suivis (Following)

Un utilisateur Twitch a la possibilité de suivre une chaîne Twitch, le principe du "suivis" est que chaque fois que la chaîne commence à diffuser du flux, les utilisateurs qui suivent cette chaîne sont prévenu via e-mail.

## 2.2 Pourquoi avoir choisi ce sujet ?

J'ai choisi ce sujet parce que je porte un réel intérêt au monde du streaming. En effet, je regarde régulièrement des streams sur des chaînes de la plateforme de streaming Twitch. N'étant jamais au courant de quel streamer est en train de diffuser à moment donné, j'ai donc pensé à une application qui notifie et informe l'utilisateur des différents streams en cours.

## 2.3 Pourquoi une application de bureau ?

Au départ, je voulais créer une application mobile, ce qui augmente l'intérêt de l'application mais, le temps laissé à disposition n'est pas assez conséquent et donc étant novice dans la programmation mobile j'ai décidé de créer une application de bureau que je pourrai peut-être porter plus tard sur mobile.

---

2. Video On Demand

## 2.4 Pourquoi C# ?

J'ai choisi de développer mon application en C# car c'est le langage que je connais le mieux actuellement, et je pense également que c'est le langage qui est le plus adapté pour l'application à réaliser car elle est sur Windows.

# 3 Analyse de l'existant

## 3.1 Existant

Étant donné que le streaming est un monde nouveau il y a peu d'application sur ce sujet, en effet les seules applications sont :

- **Twitch** : Application mobile officielle de la plateforme de streaming Twitch
- **Twitch now** : Application Google Chrome
- **Twitch Notifier** : Application Google Chrome

## 3.2 Critique de l'existant

**Twitch** : Cette application sert à accéder à la plateforme de streaming Twitch. En effet, c'est l'application officielle, il y a donc la possibilité de visionner n'importe quelles chaînes Twitch, la possibilité de se connecter à la plateforme pour voir toutes les chaînes que l'utilisateur suit.

**Twitch now** : Cette application permet à l'utilisateur de naviguer dans les différents streams de Twitch, c'est une application Google Chrome de Twitch, elle permet de naviguer dans Twitch sans aller sur le site. Le système de suivi est celui de Twitch, c'est-à-dire que lorsque dans l'application on suit une chaîne Twitch, dans le compte Twitch de l'utilisateur la chaîne s'ajoute également dans la liste de suivi.

**Twitch notifier** : Cette application permet à l'utilisateur d'être notifié lorsqu'un stream commence. C'est une application pour Google Chrome donc inutilisable si l'utilisateur n'utilise pas Google Chrome. C'est une application qui s'approche beaucoup de celle que je réalise car on peut saisir, dans l'application, la chaîne Twitch d'un streamer et lorsque ce streamer commence à streamer, une notification s'affiche sous forme de pop-up.

La lecture des flux vidéos se fait directement sur le site Twitch.

## 4 Analyse fonctionnelle

### 4.1 Gestion des chaines

La gestion des chaines à pour but de gérer les différentes chaîne. Elle se compose comme ceci : Liste des streams en ligne et hors-ligne, détail d'une chaîne Twitch.

La liste des streams en ligne affiche les streams que l'utilisateur suit et qui sont actuellement en ligne, la liste des stream hors-ligne affiche les streams que l'utilisateur suit et qui sont actuellement hors-ligne. Les différentes options de configuration sur chaque stream sont accessible depuis ces liste.

Le détail d'une chaîne Twitch consiste à afficher toutes les informations concernant une chaîne Twitch, en effet, c'est depuis là que l'utilisateur peut visualiser le stream de la chaîne, également la possibilité d'afficher le chat de la chaîne, toutes les fonctions de bases de Twitch pour une chaîne est repris ici.

### 4.2 Recherche d'une chaîne

Il y a différente manière de rechercher une chaîne sur Twitch :

- Recherche par nom de chaîne
- Recherche par jeux
- Recherche par popularité

L'application reprend les mêmes filtres que le site Twitch.

### 4.3 Ajout d'une chaîne

L'ajout d'une chaîne Twitch se fait directement depuis le détail d'une chaîne ou depuis les résultats d'une recherche. Le terme "Ajout" signifie que l'on ajoute la chaîne à la liste des Suivis de l'utilisateur courant.

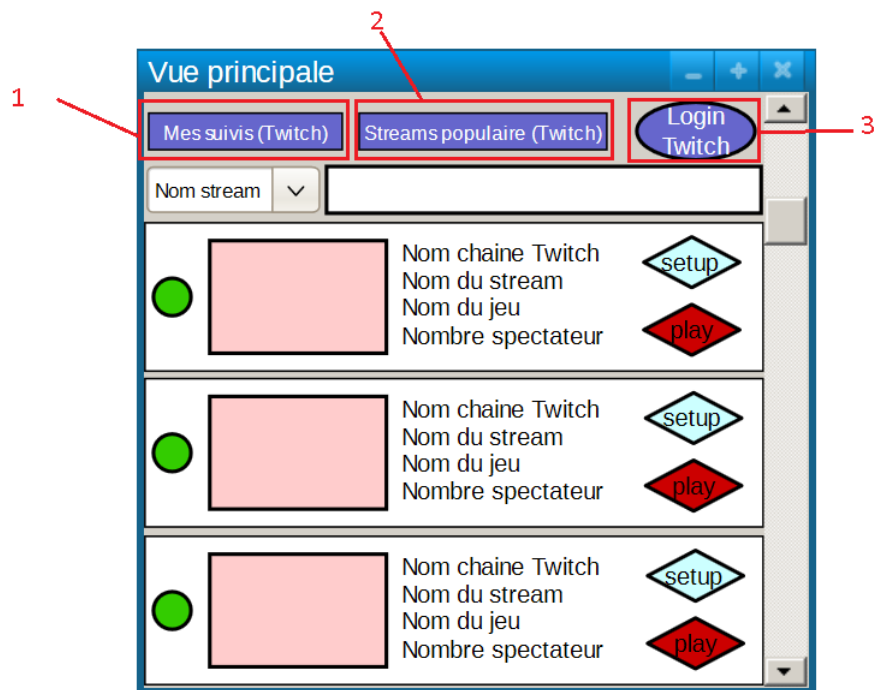
### 4.4 Gestion des notifications

La gestion des notifications à pour but de gérer les notifications d'un stream lorsqu'il démarre.

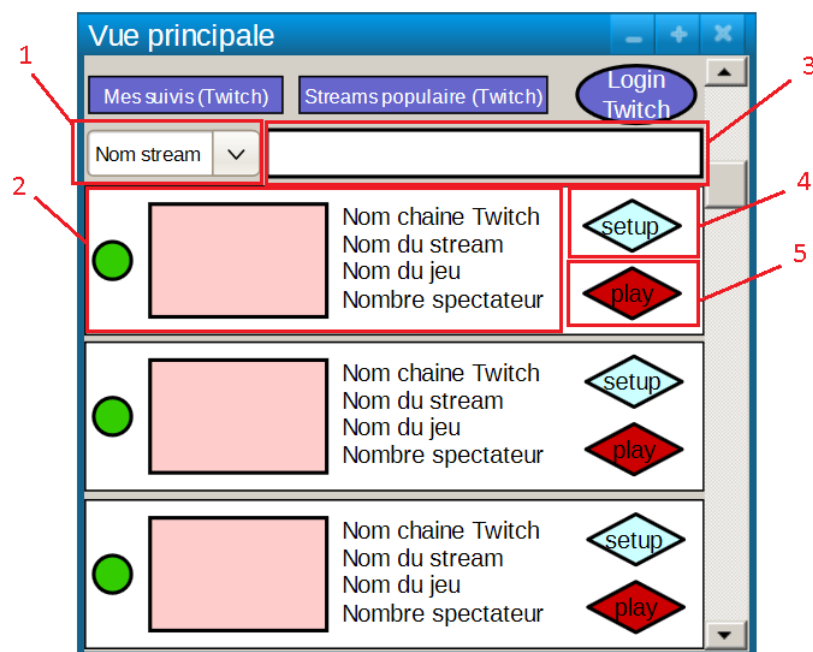
### 4.5 Interface homme-machine

#### 4.5.1 Vue principale

La vue principale permet à l'utilisateur connecté à Twitch de s'informer des différents streams en cours. Elle permet également de rechercher une chaîne Twitch pour la regarder ou pour la suivre. C'est depuis cette vue que l'on peut accéder à la configuration des notifications de chaque chaîne suivis.



1. Affiche les suivis du comptes Twitch actuellement connecté
2. Affiche les streams populaire actuel de Twitch
3. Permet de se connecter à Twitch



1. Paramètre sur lequel la recherche va être effectuée
2. Information concernant le stream
3. Champ de saisie pour la recherche



4. Bouton pour accéder à la configuration de la chaîne
5. Bouton pour lire le stream depuis l'application

## 5 Analyse organique

### 5.1 Diagramme de classe

### 5.2 API Twitch (Twitch.Net)

L'API Twitch sert à récupérer les différentes informations d'un flux vidéo sur Twitch. Elle utilise un lien URL pour retrouver toutes les informations d'un stream :

`https://api.twitch.tv/kraken/channels/[nom de la chaîne Twitch]`

Toutes les informations que l'API appelle se fait depuis cette page qui est formatée en JSON<sup>3</sup>.

L'API est composé de cette manière :

- Channels : Informations concernant une chaîne Twitch
- Chat : Informations concernant le *chat* de la chaîne Twitch
- Follows : Informations concernant les "suiveurs" de la chaîne Twitch
- Games : Informations concernant le jeu en cours sur la chaîne Twitch
- Search : Recherche des streams et des jeux
- Streams : Informations concernant le stream
- Subscriptions : Informations concernant les abonnements de la chaîne Twitch
- Teams : Informations concernant une team sur Twitch
- Users : Informations concernant un utilisateur
- Videos : Informations concernant les vidéos de la chaîne Twitch

#### 5.2.1 OAuth 2.0

Pour l'authentification d'un utilisateur Twitch sur une application externe, l'API utilise le protocole **OAuth 2.0**, en effet, ce protocole permet d'obtenir un accès limité à un service via HTTP par le biais d'une autorisation. La demande d'accès est demandée par le "client", dans notre cas, l'application est le client.

OAuth2 définit 4 rôles :

- Détenteur des données (Nous-même)
- Serveur de ressources (Twitch)
- Le client (Twitch Alert)
- Serveur d'autorisation (Twitch)

#### Token(jeton)

Lorsque le client fait une demande d'authentification, le serveur d'autorisation délivre un token. Un token permet au serveur de ressources d'autoriser la mise à disposition des données d'un utilisateur. Un token a une durée de vie limitée qui est définie par le serveur qui délivre les tokens. Un token doit rester le plus confidentiel possible, même l'utilisateur des ressources ne voit pas le token.

#### Scope(portée)

Le scope est un paramètre qui sert à définir des droits sur le token. Le serveur d'autorisation propose une liste de scope disponible puis, lors de la demande d'un token, il faut spécifier les scopes à mettre en place.

#### Type d'autorisation

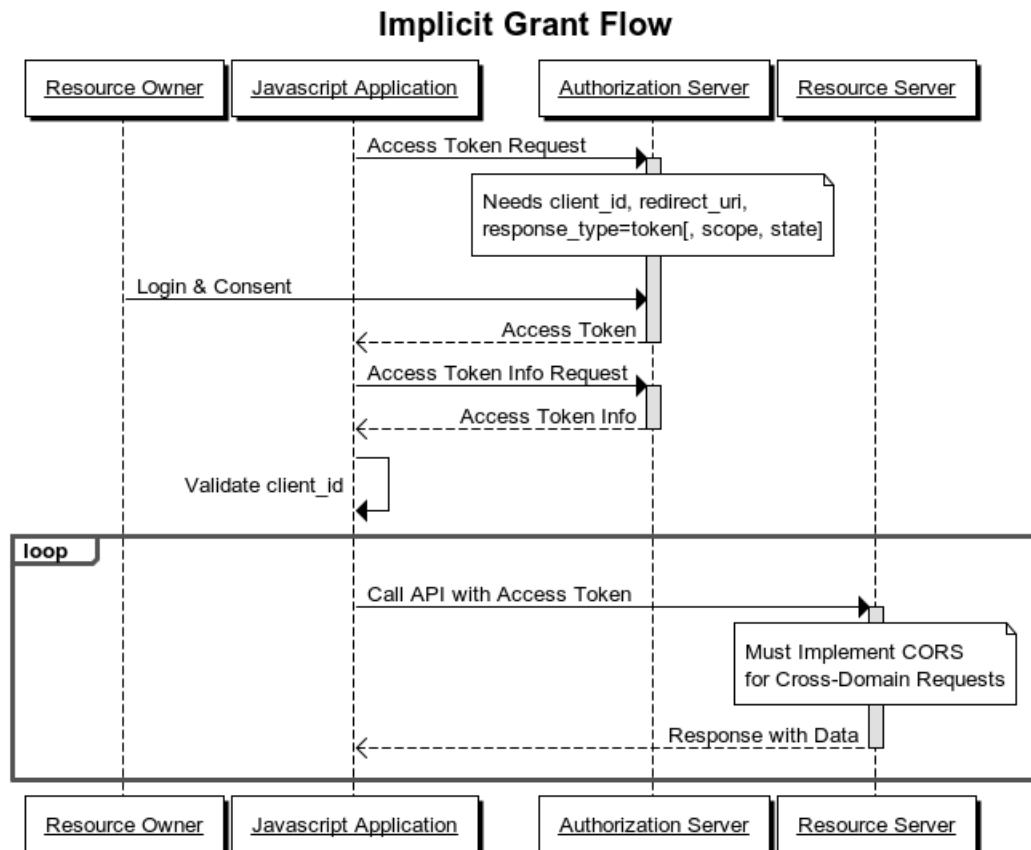
Il existe deux types d'autorisation, celui qui nous intéresse est : L'autorisation implicite (*Implicit Grant*) L'autorisation implicite s'utilise quand l'application se trouve coté client, ce qui est notre cas. La demande se fait de la sorte :

1. L'application souhaite accéder aux données Twitch de votre compte
2. Redirigé dans un navigateur web par l'application vers le serveur d'autorisation
3. Si l'accès est autorisé, le serveur d'autorisation délivre le token et on peut le voir dans l'url.

---

3. JavaScript Object Notation

4. On utilise le token pour faire des appels à l'API de Twitch.



### 5.2.2 Site d'autorisation

Lorsqu'un utilisateur veut se connecter et ainsi autoriser le compte à utiliser mon application, celui-ci est redirigé vers un site internet. Ce site internet que l'on a accès via : <http://froelicher.github.io/TwitchAlert/>. Ce site a été mis en ligne par moi-même mais le code a été pris de l'exemple donné par Twitch dans la documentation de OAuth 2.0. C'est également sur ce site que l'on récupère le token dans l'URL.

### 5.2.3 cURL

Client **URL Request Library** est une interface en ligne de commande. Elle permet de récupérer des données d'une ressource accessible par réseau. On désigne la ressource à partir d'une URL.

```
curl -H 'Accept: application/vnd.twitchtv.v3+json' \
-X GET https://api.twitch.tv/kraken/channels/test_user1/follows
```

Ci-dessus un exemple de requête envoyée pour récupérer les données d'une ressource.

Dans notre cas, nous envoyons une requête de type "GET" pour récupérer par exemple les informations d'une chaîne puis on reçoit les données.

Les différents type de requêtes :

- GET
- POST
- PUT
- DELETE

## 6 Tests

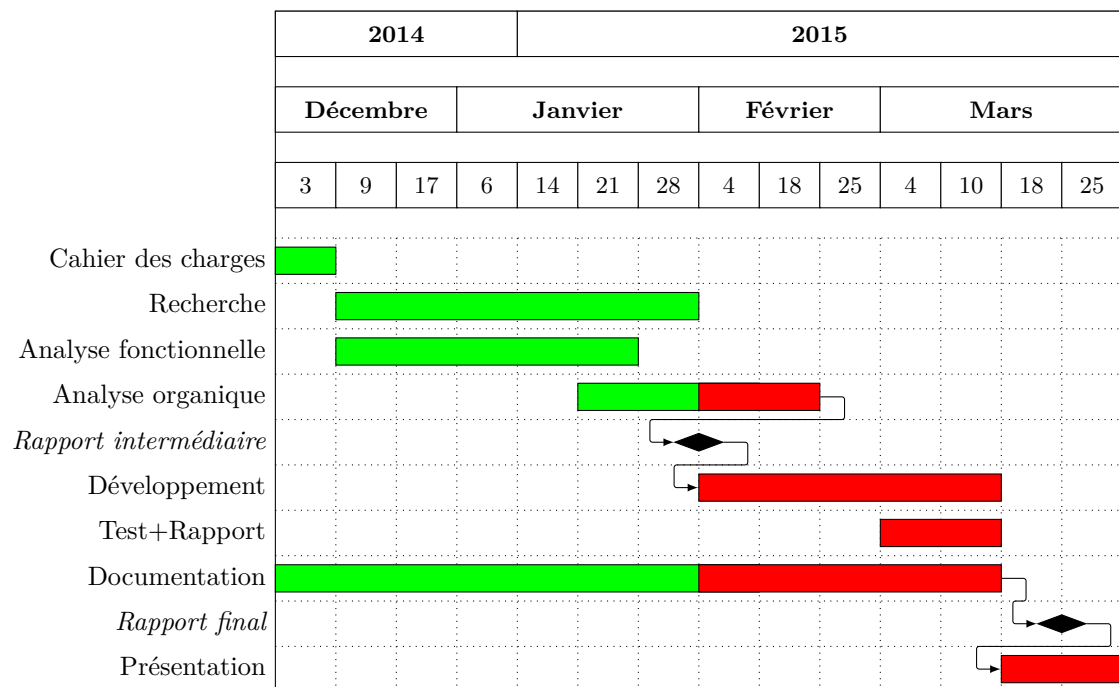
## 7 Conclusion

## 8 Glossaire

## 9 Annexes

### 9.1 Planification

#### 9.1.1 Planification prévisionnelle



## 9.1.2 Planning réel

