

# INF3005 – Programmation web avancée

## Flask – partie 3

Jacques Berger

# Objectifs

Paramètres d'URL

Gestion de page introuvable

Cookies

# Prérequis

Flask

HTTP

# Paramètres d'URL

Une route peut contenir un élément variable

Ex.

`https://twitter.com/jacquesberger`

La partie 'jacquesberger' est un paramètre dans l'URL représentant un nom d'utilisateur

# Paramètres d'URL

Lorsqu'on déclare une route avec un élément variable, l'élément variable est passé en paramètre à la fonction associée à la route

# Paramètres d'URL

```
@app.route('/artiste/<identifiant>')
def artist_page(identifiant):
    artist = get_db().get_artist(identifiant)
    ...
```

# Page 404

Lorsqu'on demande une route qui n'existe, ou qu'un paramètre d'URL fait référence à une ressource qui n'existe pas, on peut générer une page d'erreur

# Page 404

404 Not found

404 est le code HTTP pour une ressource introuvable

Lorsqu'on ne trouve pas la route, ou les paramètres font référence à quelque chose d'inexistant, le server doit retourner un code 404



# Page 404

En plus du code, on peut retourner une page pour représenter l'erreur de façon esthétique

Ex.

`https://twitter.com/jacquesTTTTTberger`

# Page 404

## Le render\_template permet tout ça

```
@app.route('/artiste/<identifiant>')
def artist_page(identifiant):
    artist = get_db().get_artist(identifiant)
    if artist is None:
        return render_template('404.html'), 404
    else:
        ...
```

# Réponse

La réponse à une requête est automatiquement gérée par Flask lorsqu'on fait un return dans une fonction de route

# Réponse

Par contre, dans certains cas, on doit pouvoir changer certaines données dans la réponse

On doit d'abord créer un objet de réponse avec la fonction `make_response`

# Réponse

```
response = make_response(render_template('artiste.html',  
                                         artist=artist))
```

# Réponse

Ensuite, on peut accéder à des propriétés et fonctions de l'objet réponse

En effectuant un return de l'objet réponse, la réponse sera envoyée au fureteur

# Cookies

Un cookie est une variable stockée au niveau du fureteur

Il s'agit d'un concept relatif à HTTP

Une fois défini, le cookie est envoyé au serveur à chaque requête

# Cookies

Les cookies sont associés à un nom de domaine

Toutes les requêtes faites à ce nom de domaine, par un fureteur en particulier, recevront tous les cookies définis sur le site



# Cookies

Les cookies sont stockés dans un fichier texte par le fureteur, sans aucune encryption

Ne jamais y placer des données confidentielles

# Cookies

Les cookies peuvent être déclarés par le frontend en Javascript, ou par le backend

# Cookies

Déclarer un cookie avec Flask (avec l'objet response) :

```
response = make_response(  
    render_template('artiste.html',  
                    artist=artist))  
response.set_cookie("last", identifier)  
return response
```

# Cookies

Le cookie peut ensuite être lu dans n'importe quelle route avec l'objet request

```
last_id = request.cookies.get('last')
```

# Plus loin...

## Cookies

<http://flask.pocoo.org/docs/0.12/quickstart/#cookies>

## Pages d'erreur

<http://flask.pocoo.org/docs/0.12/patterns/errorpages/>