# Technologie internet :

# WEB API

## Controller User :

### GetUser()

**Action :**

A pour but de récupérer l’ensemble des utilisateurs de la base de données.

**Retourne :**

Retourne une liste contenant l’ensemble des utilisateurs.

**Exception :**

Retourne Unauthorized si le token n’est plus valdie

### GetUser(int id)

**Action :**

A pour but de récupérer un utilisateur particulier.

**Retourne :**

Retourne un utilisateur.

**Exception :**

Retourne NotFound si aucun n’utilisateur trouvé

Retourne Unauthorized si le token n’est plus valdie

### PutUser(int id, User user)

**Action :**

A pour but de modifier les données d’un utilisateur particulier

**Retourne :**

Ne retourne rien.

**Exception :**

Retourne NotFound si aucun n’utilisateur trouvé

Retourne BadRequest si l’id et l’user id ne correspond pas

Retourn Nocontent si l’user ne contient pas d’information

Retourne Unauthorized si le token n’est plus valdie

### PostUser(User user)

**Action :**

A pour but d’ajouter un utilisateur dans la base de données.

**Retourne :**

Retourne l’utilisateur ajouté

**Exception :**

Retourne BadRequest si l’id et l’user id ne correspond pas

Retourne Unauthorized si le token n’est plus valdie

### DeleteUser(int id)

**Action :**

A pour but de supprimer un utilisateur particulier de la base de données.

**Retourne :**

Ne retourne rien

**Exception :**

Retourne NotFound si aucun n’utilisateur trouvé

Retourn Nocontent si l’user ne contient pas d’information

Retourne Unauthorized si le token n’est plus valdie

### UserExist(int id)

**Action :**

A pour but de vérifier qu’un utilisateur particulier existe dans la base de données.

**Retourne :**

Retourne vraie si trouvé et faux si non trouvé

## Controller Show :

### GetShow()

**Action :**

A pour but de récupérer l’ensemble des shows de la base de données.

**Retourne :**

Retourne une liste contenant l’ensemble des shows.

### GetShow(int id)

**Action :**

A pour but de récupérer un show particulier.

**Retourne :**

Retourne un show.

**Exception :**

Retourne NotFound si aucun show trouvé

### PutShow(int id, Show show)

**Action :**

A pour but de modifier les données d’un show particulier

**Retourne :**

Ne retourne rien.

**Exception :**

Retourne NotFound si aucun show trouvé

Retourne BadRequest si l’id et le show id ne correspond pas

Retourn Nocontent si le show ne contient pas d’information

### PostShow(Show show)

**Action :**

A pour but d’ajouter un show dans la base de données.

**Retourne :**

Retourne le show ajouté

**Exception :**

Retourne BadRequest si l’id et le show id ne correspond pas

### DeleteShow(int id)

**Action :**

A pour but de supprimer un show particulier de la base de données.

**Retourne :**

Ne retourne rien

**Exception :**

Retourne NotFound si aucun show trouvé

Retourn Nocontent si le show ne contient pas d’information

### ShowExist(int id)

**Action :**

A pour but de vérifier qu’un utilisateur particulier existe dans la base de données.

**Retourne :**

Retourne vraie si trouvé et faux si non trouvé

## Controller Episode :

### GetEpisode()

**Action :**

A pour but de récupérer l’ensemble des épisodes de la base de données.

**Retourne :**

Retourne une liste contenant l’ensemble des épisodes.

Retourne Unauthorized si le token n’est plus valdie

### GetEpisode(int id)

**Action :**

A pour but de récupérer un épisode particulier.

**Retourne :**

Retourne un épisode.

**Exception :**

Retourne NotFound si aucun épisode trouvé

Retourne Unauthorized si le token n’est plus valdie

### PutEpisode(int id, Episode episode)

**Action :**

A pour but de modifier les données d’un épisode particulier

**Retourne :**

Ne retourne rien.

**Exception :**

Retourne NotFound si aucun épisode trouvé

Retourne BadRequest si l’id et l’épisode id ne correspond pas

Retourn Nocontent si l’épisode ne contient pas d’information

Retourne Unauthorized si le token n’est plus valdie

### PostEpisode(Episode episode)

**Action :**

A pour but d’ajouter un épisode dans la base de données.

**Retourne :**

Retourne l’épisode ajouté

**Exception :**

Retourne BadRequest si l’id et l’épisode id ne correspond pas

Retourne Unauthorized si le token n’est plus valdie

### DeleteEpisode(int id)

**Action :**

A pour but de supprimer un épisode particulier de la base de données.

**Retourne :**

Ne retourne rien

**Exception :**

Retourne NotFound si aucun épisode trouvé

Retourn Nocontent si l’épisode ne contient pas d’information

Retourne Unauthorized si le token n’est plus valdie

### EpisodeExist(int id)

**Action :**

A pour but de vérifier qu’un épisode particulier existe dans la base de données.

**Retourne :**

Retourne vraie si trouvé et faux si non trouvé

## Controller Connection User :

### EpisodeExist(int id)

**Action :**

A pour but de permettre la connexion d’un user en vérifiant ses informations dans la base de données.

**Retourne :**

Retourne l’id de l’utilisateur si trouvé sinon retourne 0 ;

## Controller TokenWallets :

### GetTokenWallet(string name, string password)

**Action :**

A pour but de récupérer l’ensemble des tokenWallets de la base de données.

**Retourne :**

Retourne une liste contenant l’ensemble des tokenWallets.

### GetTokenWallet(int id)

**Action :**

A pour but de récupérer un tokenWallet particulier.

**Retourne :**

Retourne un tokenWallet.

**Exception :**

Retourne NotFound si aucun épisode trouvé

### PutTokenWallet (int id, TokenWallet tokenWallet)

**Action :**

A pour but de modifier les données d’un tokenWallet particulier

**Retourne :**

Ne retourne rien.

**Exception :**

Retourne NotFound si aucun épisode trouvé

Retourne BadRequest si l’id et le tokenWallet id ne correspond pas

Retourn Nocontent si le tokenWallet ne contient pas d’information

### PostTokenWallet(TokenWallet tokenWallet)

**Action :**

A pour but d’ajouter un tokenWallet dans la base de données.

**Retourne :**

Retourne le tokenWallet ajouté

**Exception :**

Retourne BadRequest si l’id et le tokenWallet id ne correspond pas

### DeleteTokenWallet(int id)

**Action :**

A pour but de supprimer un tokenWallet particulier de la base de données.

**Retourne :**

Ne retourne rien

**Exception :**

Retourne NotFound si aucun épisode trouvé

Retourn Nocontent si l’épisode ne contient pas d’information

### EpisodeExist(int id)

**Action :**

A pour but de vérifier qu’un tokenWallet particulier existe dans la base de données.

**Retourne :**

Retourne vraie si trouvé et faux si non trouvé