



PROGRAMMATION MOBILE  
**ANDROID**

---

PROJET

## TABLE DES MATIERES

Description du projet.....	3
Fonctionnalités minimales attendues .....	3
Compte utilisateur.....	3
Liste des peripheriques.....	3
Bonus .....	3
Barème de notation.....	4
Documentation de l' API.....	4
Classe Kotlin API.....	4
Creation d' un compte utilisateur.....	4
Connexion .....	5
Liste des maisons .....	6
Donner acces a un tiers a une maison.....	6
Supprimer l' acces d' un tiers a une maison .....	7
Lister les peripheriques d' une maison.....	8
Transmettre une commande a un peripherique.....	9

## DESCRIPTION DU PROJET

Vous venez d'acquérir un chalet moderne équipé d'objets connectés : volets, porte de garage, éclairages. Vous avez testé plusieurs solutions domotique pour piloter tous ces équipements, mais aucun ne vous convient parfaitement. Aussi, vous avez pris la décision de développer votre propre application mobile.

La centrale domotique à laquelle sont raccordés tous les équipements pilotables de la maison est déjà en place et propose une API web décrite plus loin dans le document. Une maquette virtuelle permettant de visualiser en temps réel l'état de la maison est disponible à cette adresse :

<https://polyhome.lesmoulinstudev.com?houseId=<houseId>>

A partir des possibilités fournies par cette API, réalisez l'application de domotique de vos rêves !

Notes : le meilleur projet sera exposé pour les journées portes ouvertes.

## FONCTIONNALITÉS MINIMALES ATTENDUES

### COMPTÉ UTILISATEUR

L'application devra permettre de créer un compte qui sera utiliser pour vous identifier auprès de l'API.

Une fois le compte créé, l'application proposera une interface de connexion. Afin de simplifier l'utilisation de l'application, cette dernière mémorisera localement le token d'authentification retourné par l'API au moment de la connexion.

### LISTE DES PÉRIPHÉRIQUES

L'application sera en mesure de lister les périphériques présents dans la maison et de leurs transmettre des commandes.

### BONUS

Il est recommandé d'ajouter des fonctionnalités bonus qui simplifieront la vie de l'utilisateur. Par exemple, pouvoir piloter plusieurs équipements d'une seule action ou permettre à d'autres utilisateurs de piloter votre maison.

## BAREME DE NOTATION

Evaluation des fonctionnalités	
Création de compte	2
Connexion	2
Lister les périphériques de la maison	4
Transmettre des commandes	2
Fonctionnalités bonus	7
Ergonomie (utilisation aisée et intuitive)	2
Aspect visuel	1
Malus	
Mauvaise qualité du code	-10
Retard de rendu	-20

## DOCUMENTATION DE L'API

### CLASSE KOTLIN API

Vous pouvez utiliser la classe Api ci-dessous pour interroger l'API :

<https://www.lamarmotte.info/wp-content/uploads/2024/11/Api.zip>

### CREATION D'UN COMPTE UTILISATEUR

Réquête client	
URL	<a href="https://polyhome.lesmoulinsdudev.com/api/users/register">https://polyhome.lesmoulinsdudev.com/api/users/register</a>
Méthode	POST
Content-Type	application/json
Body	{         "login": string         "password": string     }
Réponse serveur	
Code de réponse	200 : Le compte a bien été créé 400 : Les données fournies sont incorrectes 409 : Le login est déjà utilisé par un autre compte 500 : Une erreur s'est produite au niveau du serveur

## LISTE DES COMPTES CREEES

<b>Réquête client</b>	
URL	<a href="https://polyhome.lesmoulinstudev.com/api/users">https://polyhome.lesmoulinstudev.com/api/users</a>
Méthode	GET
<b>Réponse serveur</b>	
Code de réponse	200 : La liste des utilisateurs a bien été retournée 500 : Une erreur s'est produite au niveau du serveur

## CONNEXION

<b>Réquête client</b>	
URL	<a href="https://polyhome.lesmoulinstudev.com/api/users/auth">https://polyhome.lesmoulinstudev.com/api/users/auth</a>
Méthode	POST
Content-Type	application/json
Body	<pre>{   "login": string   "password": string }</pre>
<b>Réponse serveur</b>	
Code de réponse	200 : Connexion réussie 400 : Les données fournies sont incorrectes 404 : Aucun utilisateur ne correspond aux identifiants donnés 500 : Une erreur s'est produite au niveau du serveur
Content-Type	application/json
Body	<pre>{   "token": string }</pre>

## LISTE DES MAISONS

Réquête client	
URL	<a href="https://polyhome.lesmoulinstudev.com/api/houses">https://polyhome.lesmoulinstudev.com/api/houses</a>
Méthode	GET
Authorization	Bearer < <i>token</i> >
Réponse serveur	
Code de réponse	<p>200 : Requête acceptée          403 : Accès interdit (token invalide)          500 : Une erreur s'est produite au niveau du serveur</p>
Content-Type	application/json
Body	[ { "houseId": integer, "owner": boolean }, { ... }, ... ]

## DONNER ACCÈS À UN TIERS À UNE MAISON

Réquête client	
URL	<a href="https://polyhome.lesmoulinstudev.com/api/houses/&lt;houseId&gt;/users">https://polyhome.lesmoulinstudev.com/api/houses/&lt;houseId&gt;/users</a>
Méthode	POST
Authorization	Bearer < <i>token</i> >
Content-Type	application/json
Body	{ "userLogin": string }
Réponse serveur	
Code de réponse	<p>200 : Accès accordé          400 : Les données fournies sont incorrectes          403 : Accès interdit (token invalide ou ne correspondant pas au propriétaire de la maison)          409 : L'utilisateur est déjà associé à la maison          500 : Une erreur s'est produite au niveau du serveur</p>

## SUPPRIMER L'ACCÈS D'UN TIERS À UNE MAISON

Réquête client	
URL	<code>https://polyhome.lesmoulinsdudev.com/api/houses/&lt;houseId&gt;/users</code>
Méthode	DELETE
Authorization	Bearer <code>&lt;token&gt;</code>
Content-Type	application/json
Body	<pre>{     "userLogin": string }</pre>
Réponse serveur	
Code de réponse	<p>200 : Suppression réalisée</p> <p>400 : Les données fournies sont incorrectes</p> <p>403 : Accès interdit (token invalide ou ne correspondant pas au propriétaire de la maison)</p> <p>500 : Une erreur s'est produite au niveau du serveur</p>

## LISTER LES UTILISATEURS ASSOCIES À UNE MAISON

Réquête client	
URL	<code>https://polyhome.lesmoulinsdudev.com/api/houses/&lt;houseId&gt;/users</code>
Méthode	GET
Authorization	Bearer <code>&lt;token&gt;</code>
Réponse serveur	
Code de réponse	<p>200 : La liste des utilisateurs a bien été envoyée</p> <p>400 : Les données fournies sont incorrectes</p> <p>403 : Accès interdit (token invalide ou ne correspondant pas au Propriétaire ou à un invité de la maison)</p> <p>500 : Une erreur s'est produite au niveau du serveur</p>
Content-Type	application/json
Body	<pre>[     {         "userLogin": string,         "owner": number     },     {         "userLogin": string,         "owner": number     },     ... ]</pre>

## LISTER LES PERIPHERIQUES D'UNE MAISON

Attention : au moins une instance de la maison doit être ouverte dans un navigateur !

<https://polyhome.lesmoulinstudev.com?houseId=<houseId>>

Réquête client	
URL	<a href="https://polyhome.lesmoulinstudev.com/api/houses/&lt;houseId&gt;/devices">https://polyhome.lesmoulinstudev.com/api/houses/&lt;houseId&gt;/devices</a>
Méthode	GET
Authorization	Bearer <token>
Réponse serveur	
Code de réponse	<p>200 : Requête acceptée  400 : Les données fournies sont incorrectes  403 : Accès interdit (token invalide ou ne correspondant pas au propriétaire de la maison ou à un tiers ayant accès)  500 : Une erreur s'est produite au niveau du serveur</p>
Content-Type	application/json
Body	<pre>{   devices : [     {       "id": string,       "type": string,       "availableCommands": [string, string, ...],       "opening": number,     },     {       "id": string,       "type": string,       "availableCommands": [string, string, ...],       "power": number,     },     ...   ] }</pre>

## TRANSMETTRE UNE COMMANDE A UN PERIPHERIQUE

Réquête client	
URL	<code>https://polyhome.lesmoulinsdudev.com/api/houses/&lt;houseId&gt;/devices/&lt;deviceId&gt;/command</code>
Méthode	POST
Authorization	Bearer <code>&lt;token&gt;</code>
Réponse serveur	
Code de réponse	<p>200 : Requête acceptée          403 : Accès interdit (token invalide ou ne correspondant pas au propriétaire de la maison ou à un tiers ayant accès)          500 : Une erreur s'est produite au niveau du serveur</p>
Content-Type	application/json
Body	<pre>{     "command" : string }</pre>