

BIRDY

**Sorbonne Université 2020 - UE LU3IN017 - Anes BOUZOUAOUI & Chems Eddine BENSAFIA**

|  |  |
| --- | --- |
| Service 1 : « Création d’utilisateurs » | |
| Nom du web service | createUser |
| URL du web service | POST /api/users/register |
| Description du service | Permet de créer un nouvel utilisateur, et l’ajouter à la base de données |
| Paramètres en entrée | Dans le body de la requete et sous format JSON : username, firstName, lastName, email, password |
| Format de sortie | JSON contenant les mêmes paramètres d’entrée avec un status code de 201 (created) |
| Exemple de sortie | 1. {status: (200), message: ‘Welcome’ 2. {status: (409), message: 'User already exists'} 3. {status: (400), message: 'Bad inputs'} 4. {status: (500), message: 'database problem'} |
| Erreurs possibles | tentative de créer un utilisateur qui existe, la clef primaire étant l'email et username  inputs error par exemple un mot de passe trop court ou champ adressse mail non valide  problème de base de donnée |
| Avancement du Service | PRET |
| Classes/Fichiers JavaScript | /routes/api/users.js |
| Informations additionnelles | Validation des informations de l’utilisateur se fait a l’aide du module populaire « Joi » |

|  |  |
| --- | --- |
| Service 2 : « connexion d’un utilisateur » | |
| Nom du web service | loginUser |
| URL du web service | POST /api/users/login |
| Description du service | Permet à un utilisateur de se connecter, et de commencer une nouvelle session |
| Paramètres en entrée | Dans le body de la requete et sous format JSON : email et password |
| Format de sortie | JSON contenant les mêmes paramètres d’entrée avec un status code de 201 (created) |
| Exemples de sortie | 1. {status: (200), message: ‘Welcome’ 2. {status: (404), message: 'no matching email'} 3. {status: (401), message: 'Bad password'} 4. {status: (500), message: 'database problem'} |
| Erreurs possibles | Utilisateur inconnu, mauvais mot de passe  problème de base de donnée |
| Avancement du Service | PRET |
| Classes/Fichiers JavaScript | /routes/api/users.js |
| Informations additionnelles | Gestion de la sécurité des mots de passe a l’aide du module « bcrypt» |

|  |  |
| --- | --- |
| Service 3 : « Logout » | |
| Nom du web service | logout |
| URL du web service | POST /api/users/logout |
| Description du service | Permet à un utilisateur de se déconnecter |
| Paramètres en entrée | Néant |
| Format de sortie | Status code indiquant que c’est bien déconnecté |
| Exemples de sortie | 1. {status: (204), message: ‘Welcome’ 2. {status: (401), message: 'Token is not valid'} 3. {status: (401), message: ‘No token, authorization denied’} |
| Erreurs possibles | Utilisateur non authentifié |
| Avancement du Service | PRET |
| Classes/Fichiers JavaScript | /routes/api/users.js  /middleware/auth.js |
| Informations additionnelles | Néant |

|  |  |
| --- | --- |
| Service 4 : « Ajouter un ami » | |
| Nom du web service | addFriend |
| URL du web service | POST /api/friends/add\_friend |
| Description du service | Permet à un utilisateur connecté d’ajouter un ami |
| Paramètres en entrée | Le username de l’ami |
| Format de sortie | Status code indiquant que c’est bien déconnecté |
| Exemples de sortie | 1. {status: (201), message: ‘Friend added’ 2. {status: (401), message: 'Token is not valid'} 3. {status: (401), message: ‘No token, authorization denied’} |
| Erreurs possibles | Utilisateur non authentifié  Probleme au niveau de la base de données |
| Avancement du Service | PRET |
| Classes/Fichiers JavaScript | /routes/api/ friends.js  /middleware/auth.js |
| Informations additionnelles | Néant |

|  |  |
| --- | --- |
| Service 5: « Créer ou modifier le profile » | |
| Nom du web service | Create/UpdateUserProfile |
| URL du web service | POST /api/profils |
| Description du service | Permet de créer son profil ou de le mettre a jour |
| Paramètres en entrée | location, bio, status, githubusername, intrests |
| Format de sortie | Status code indiquant que c’est bien déconnecté |
| Exemples de sortie | 1. {status: (401), message: 'Token is not valid'} 2. {status: (401), message: ‘No token, authorization denied’} |
| Erreurs possibles | Utilisateur non authentifié  Probleme au niveau de la base de données  Validation des champs non validé |
| Avancement du Service | PRET |
| Classes/Fichiers JavaScript | /routes/api/ profils.js  /middleware/auth.js |
| Informations additionnelles | Néant |

|  |  |
| --- | --- |
| Service 6: « Delete profile,user & post» | |
| Nom du web service | deleteUser |
| URL du web service | DELETE api/profils |
| Description du service | Permet d’effectuer une recherche d’un profil et de le supprimer ainsi que ses posts |
| Paramètres en entrée | username |
| Format de sortie | Message indiquant que le profil indiqué a bien été supprimé |
| Exemples de sortie | 1. {message: ‘User deleted’} |
| Erreurs possibles | Problème au niveau de la base de données  Server error |
| Avancement du Service | PRET |
| Classes/Fichiers JavaScript | /routes/api/ profils.js |
| Informations additionnelles | Néant |

|  |  |
| --- | --- |
| Service 7: « Créer un post» | |
| Nom du web service | CreatePost |
| URL du web service | POST api/posts |
| Description du service | Permet de créer un post (publication) |
| Paramètres en entrée | Texte de la publication |
| Format de sortie | Status code indiquant que c’est bien déconnecté |
| Exemples de sortie | 1. {status: (401), message: 'Token is not valid'} 2. {status: (401), message: ‘No token, authorization denied’} 3. {status: (201), message: ‘created’} |
| Erreurs possibles | Utilisateur non authentifié  Probleme au niveau de la base de données  Non validation des champs |
| Avancement du Service | PRET |
| Classes/Fichiers JavaScript | /routes/api/ posts.js  /middleware/auth.js |
| Informations additionnelles | Néant |

|  |  |
| --- | --- |
| Service 8: « Get all posts» | |
| Nom du web service | GetallPosts |
| URL du web service | GET api/posts |
| Description du service | Permet de lister toute les publication de l’utilisateur |
| Paramètres en entrée | Néant |
| Format de sortie | Les publications sous format json |
| Exemples de sortie | 1. res.json(posts) |
| Erreurs possibles | Utilisateur non authentifié  Probleme au niveau de la base de données  Non validation des champs |
| Avancement du Service | PRET |
| Classes/Fichiers JavaScript | /routes/api/ posts.js  /middleware/auth.js |
| Informations additionnelles | Néant |

|  |  |
| --- | --- |
| Service 9: « Get posts by id» | |
| Nom du web service | GetPostsById |
| URL du web service | GET api/posts/:id |
| Description du service | Permet de lister toute les publication d’un utilisateur donné |
| Paramètres en entrée | Username de l’utilisateur en question |
| Format de sortie | Les publications sous format json |
| Exemples de sortie | 1. res.json(posts) |
| Erreurs possibles | Utilisateur non authentifié  Probleme au niveau de la base de données  Non validation des champs |
| Avancement du Service | PRET |
| Classes/Fichiers JavaScript | /routes/api/ posts.js  /middleware/auth.js |
| Informations additionnelles | Néant |

|  |  |
| --- | --- |
| Service 10: « Like post by id» | |
| Nom du web service | LikePostByID |
| URL du web service | PUT api/posts/like/:id |
| Description du service | Permet de liker (mettre un j’aime) sur un post |
| Paramètres en entrée | Identifiant du post |
| Format de sortie | Likes sur le post |
| Exemples de sortie | 1. res.json(post.likes) |
| Erreurs possibles | Utilisateur non authentifié  Probleme au niveau de la base de données  Non validation des champs |
| Avancement du Service | PRET |
| Classes/Fichiers JavaScript | /routes/api/ posts.js  /middleware/auth.js |
| Informations additionnelles | Néant |

|  |  |
| --- | --- |
| Service 11: « Like post by id» | |
| Nom du web service | LikePostByID |
| URL du web service | PUT api/posts/like/:id |
| Description du service | Permet de liker (mettre un j’aime) sur un post |
| Paramètres en entrée | Identifiant du post |
| Format de sortie | Likes sur le post |
| Exemples de sortie | 1. res.json(post.likes) |
| Erreurs possibles | Utilisateur non authentifié  Probleme au niveau de la base de données  Non validation des champs |
| Avancement du Service | PRET |
| Classes/Fichiers JavaScript | /routes/api/ posts.js  /middleware/auth.js |
| Informations additionnelles | Néant |

|  |  |
| --- | --- |
| Service 12: « commenter post by id» | |
| Nom du web service | LikePostByID |
| URL du web service | POST api/posts/comment/id |
| Description du service | Permet de commenter un post |
| Paramètres en entrée | Identifiant du post |
| Format de sortie | Commentaires du post |
| Exemples de sortie | 1. res.json(post.comments) |
| Erreurs possibles | Utilisateur non authentifié  Probleme au niveau de la base de données  Non validation des champs |
| Avancement du Service | PRET |
| Classes/Fichiers JavaScript | /routes/api/ posts.js  /middleware/auth.js |
| Informations additionnelles | Néant |

|  |  |
| --- | --- |
| Service 13: « Supprimer un commenter sur un post» | |
| Nom du web service | delete a comment on post |
| URL du web service | DELETE api/posts/comment/id/comment\_id |
| Description du service | Permet de supprimer un commenter d’un post |
| Paramètres en entrée | Identifiant du post + identifiant du commentaire |
| Format de sortie | Commentaires du post |
| Exemples de sortie | 1. res.json(post.comments) |
| Erreurs possibles | Utilisateur non authentifié  Probleme au niveau de la base de données  Commentaire existant |
| Avancement du Service | PRET |
| Classes/Fichiers JavaScript | /routes/api/ posts.js  /middleware/auth.js |
| Informations additionnelles | Néant |