

Documento de peticiones

Índice

1. loadLevels	2
2. newUser	3
3. login	4
4. loadLevel	5
5. storeLevel	6
6. commentLevel	7
7. rateLevel	8
8. getStatistics	9
9. getAdminData	10
10.blockLevel / unblockLevel	11
11.blockUser / unblockUser	12
12.eraseLevel	13

Listado de peticiones

Petición: loadLevels

Tipo: POST

Condición de llamada: Cuando se carga la vista home o se aplica un filtro.

Descripción: Solicita a backend la siguiente información:

- Un juego destacado.
- Un listado de juegos.

Puede opcionalmente contar con un campo de filtro (por nombre del nivel) que afectaría solamente a la propiedad **games**.

Códigos de retorno: 200 -> OK.

Listing 1: input

```
1 {  
2   "filter_text": String // Campo opcional  
3 }
```

Listing 2: output

```
1 {  
2   "return_code": Integer,  
3   "history_levels": [  
4     {  
5       "id": Integer,  
6       "img": String, // Base64 de la portada del nivel.  
7       "name": String,  
8     },  
9     ...  
10  ],  
11  "games": [  
12    {  
13      "id": Integer,  
14      "img": String,  
15      "name": String,  
16    },  
17    ...  
18  ] // Colección de juegos.  
19 }
```

Petición: newUser**Tipo:** POST**Condición de llamada:** Registro de nuevo usuario.**Descripción:** Manda la información para registrar un nuevo usuario.**Códigos de retorno:**

200 -> OK

400 -> Bad Request

601 -> Username exists

Listing 3: input

```
1 {  
2   "username": String,  
3   "password": String  
4 }
```

Listing 4: output

```
1 {  
2   "return_code": Integer  
3 }
```

Petición: login**Tipo:** POST**Condición de llamada:** Login de usuario.**Descripción:** Manda la información para acceder con un usuario.**Códigos de retorno:**

200 -> OK

400 -> Bad Request

401 -> Unauthorized

Listing 5: input

```
1 {  
2   "username": String,  
3   "password": String  
4 }
```

Listing 6: output

```
1 {  
2   "return_code": Integer,  
3   "is_admin": Bool  
4 }
```

Petición: loadLevel**Tipo:** POST**Condición de llamada:** Cargar el nivel**Descripción:** Manda la información del nivel para ser cargado**Códigos de retorno:**

200 -> OK

400 -> Bad Request

Listing 7: input

```
1 {  
2     "id_level": Integer  
3 }
```

Listing 8: output

```
1 {  
2     "return_code": Integer,  
3     "json": String,           // Phaser Level Configuration Object,  
4     "comments": [  
5         {  
6             "date": Datetime,  
7             "id_user": Integer,  
8             "comment": String  
9         },  
10        ...  
11    ],  
12    "rate": Float  
13 }
```

Petición: storeLevel**Tipo:** POST**Condición de llamada:** Cuando el usuario crea un nivel.**Descripción:** Guarda un nivel en BD.**Códigos de retorno:**

200 -> OK

400 -> Bad Request

Listing 9: input

```
1 {  
2     "name_level": String,  
3     "json": String           // Phaser Level Configuration Object  
4 }
```

Listing 10: output

```
1 {  
2     "return_code": Integer,  
3     "id_level": Integer,  
4     "json": String           // Phaser Level Configuration Object  
5 }
```

Petición: commentLevel**Tipo:** PUT**Condición de llamada:** Cuando el usuario comenta un nivel.**Descripción:** Guarda un comentario en BD.**Códigos de retorno:**

200 -> OK

400 -> Bad Request

Listing 11: input

```
1 {  
2   "id_user": Integer,  
3   "id_level": Integer,  
4   "comment": String  
5 }
```

Listing 12: output

```
1 {  
2   "return_code": Integer  
3 }
```

Petición: rateLevel**Tipo:** PUT**Condición de llamada:** Cuando el usuario puntua un nivel.**Descripción:** Puntua un nivel.**Códigos de retorno:**

200 -> OK

400 -> Bad Request

Listing 13: input

```
1 {  
2   "id_user": Integer,  
3   "id_level": Integer,  
4   "rate": Float      // Puntuación [0-5]  
5 }
```

Listing 14: output

```
1 {  
2   "return_code": Integer  
3 }
```


Petición: getStatistics**Tipo:** GET**Condición de llamada:** Cuando el usuario accede a la vista de estadísticas.**Descripción:** Devuelve las estadísticas de uso de la aplicación.**Códigos de retorno:** 200 -> OK

Listing 15: output

```
1 {  
2   "return_code": Integer,  
3   "data": [INCOMPLETO] - POR DETERMINAR  
4 }
```

Petición: getAdminData**Tipo:** GET**Condición de llamada:** Cuando el administrador accede a la vista de administración.**Descripción:** Devuelve la lista de usuarios y niveles juntos con su estado.**Códigos de retorno:** 200 -> OK

Listing 16: output

```
1 {
2   "return_code": Integer,
3   "users": [
4     {
5       "id_user": Integer,
6       "name": String,
7       "state": Boolean // False = Blocked
8     }, ...
9   ],
10  "levels": [
11    {
12      "id_level": Integer,
13      "name": String,
14      "state": Boolean // False = Blocked
15    }, ...
16  ], ...
17 ]
18 }
```

Petición: blockLevel / unblockLevel**Tipo:** PUT**Condición de llamada:** Cuando el usuario administrador bloquea/habilita un nivel.**Descripción:** Guarda un estado en BD.**Códigos de retorno:**

200 -> OK

400 -> Bad Request

Listing 17: input

```
1 {  
2   "id_level": Integer  
3 }
```

Listing 18: output

```
1 {  
2   "return_code": Integer  
3 }
```

Petición: blockUser / unblockUser**Tipo:** PUT**Condición de llamada:** Cuando el usuario administrador bloquea/habilita un usuario.**Descripción:** Guarda un estado en BD.**Códigos de retorno:**

200 -> OK

400 -> Bad Request

Listing 19: input

```
1 {  
2   "id_user": Integer  
3 }
```

Listing 20: output

```
1 {  
2   "return_code": Integer  
3 }
```

Petición: eraseLevel**Tipo:** POST**Condición de llamada:** Cuando el usuario administrador borra un nivel.**Descripción:** Manda la información del nivel a borrar.**Códigos de retorno:**

200 -> OK

400 -> Bad Request

Listing 21: input

```
1 {  
2   "id_level": Integer  
3 }
```

Listing 22: output

```
1 {  
2   "return_code": Integer  
3 }
```