API Reference

API DOCPS

API Version: 1.0.0

API del Sistema de documentación de las pruebas de software

INDEX

Security and Authentication

SECURITY SCHEMES

KEY	TYPE	DESCRIPTION

API

1. CASOS DE PRUEBA

1.1 GET /getTestcaseById/{testcaseId}

Devuelve la información del caso de prueba con el id proporcionado y una lista de sus pasos.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*testcaseId	string	Clave compuesta del caso de prueba, representada por una cadena con el formato `IdGrupo.IdProyecto.IdPlan.IdCaso`

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

1.2 POST /createTestcase

Crea un caso de prueba para el plan de pruebas indicado.

El id del plan de pruebas (*testplanId*) se representa con una cadena en el formato IdGrupo.IdProyecto.IdPlan
priority: toma los valores 1 (baja), 2 (media) o 3 (alta)

REQUEST

REQUEST BODY - application/json

```
{
  testplanId     string
  name           string
  preconditions  string
  description    string
  priority       integer  DEFAULT:1
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

1.3 POST /updateTestcase

Actualiza la información del caso de prueba con el id proporcionado

El id es la clave compuesta del caso de prueba, representada por una cadena con el formato

IdGrupo.IdProyecto.IdPlan.IdCaso

REQUEST

REQUEST BODY - application/json

```
{
  id            string
  name          string
  preconditions string
  description   string
  priority      integer  DEFAULT:1
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

1.4 POST /saveSteps

Guarda los pasos y variables pertenecientes al caso de prueba proporcionado.

El id es la clave compuesta del caso de prueba, representada por una cadena con el formato IdGrupo.IdProyecto.IdPlan.IdCaso
steps: Es un arreglo de objetos que describen los pasos. Cada objeto posee:
- *order*: Es el orden que ocupa el paso en el caso de prueba (1 si es primero, 2 si es segundo, etc.)
- *action*, *data* y *result*: Son las propiedades acción, datos y resultado de cada paso.
- *actionVariable*, *dataVariable* y *resultVariable*: Son las variables de los campos acción, datos y resultado respectivamente. Poseen nombre y una lista de valores posibles.

REQUEST

REQUEST BODY - application/json

```
{
  id      string
  steps [{
  Array of object:
    order  integer  DEFAULT:1
    action string
    actionVariable {
      name   string
      values [string]
    }
    data   string
    dataVariable {
      name   string
      values [string]
    }
    result string
    resultVariable {
      name   string
      values [string]
    }
  }]
```

}

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

1.5 DELETE /deleteTestcase/{testcaseId}

Elimina el caso de prueba con el id proporcionado y devuelve un mensaje de error en caso de no ser posible.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*testcaseId	string	Clave compuesta del caso de prueba, representada por una cadena con el formato `IdGrupo.IdProyecto.IdPlan.IdCaso`

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

2. EJECUCIONES

2.1 POST /getExecutionsForTestcase/{testcaseId}

Devuelve la lista de ejecuciones del caso de prueba con el id proporcionado.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*testcaseId	string	Clave compuesta del caso de prueba, representada por una cadena con el formato `IdGrupo.IdProyecto.IdPlan.IdCaso`

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

2.2 POST /insertExecution

Crea una ejecución para el caso de prueba indicado.

El id del caso de prueba (*testcaseId*) se representa con una cadena en el formato IdGrupo.IdProyecto.IdPlan.IdCaso

REQUEST

REQUEST BODY - application/json

```
{
  testplanId  string
  user        integer
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

2.3 POST /updateExecutionById

Actualiza la información de la ejecución con los datos proporcionados

El id es la clave compuesta de la ejecución, representada por una cadena con el formato IdGrupo.IdProyecto.IdPlan.IdCaso.IdEjecucion

status: lleva el estado de la ejecución, y debe ser Not executed, In progress, Passed o Failed

REQUEST

REQUEST BODY - application/json

```
{
  id          string
  commentary  string
  status      enum    ALLOWED:Not executed, In progress, Passed, Failed
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

2.4 DELETE /deleteExecutionById/{executionId}

Elimina la ejecución con el id proporcionado y devuelve un mensaje de error en caso de no ser posible.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*executionId	string	Clave compuesta de la ejecución, representada por una cadena con el formato `IdGrupo.IdProyecto.IdPlan.IdCaso.IdEjecucion`

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

3. GRUPOS

3.1 GET /getGroupById/{groupId}

Devuelve el grupo con el id proporcionado

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*groupId	string	Id del grupo

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

3.2 POST /searchGroups

Devuelve una lista de grupos a los que pertenece el usuario con el id provisto, que cumplen con los criterios provistos

REQUEST

REQUEST BODY - application/json

```
{
  userId  string
  name    string
  status  enum    ALLOWED:active, inactive
  role    string
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

3.3 POST /createGroup

Crea un grupo con el nombre provisto

REQUEST

REQUEST BODY - application/json

```
{
  name  string
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

3.4 GET /getGroupAndMembersById/{groupId}

Devuelve la lista de los miembros del grupo con el id proporcionado

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*groupId	string	Id del grupo

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

3.5 POST /updateGroup

Actualiza la información del grupo con los datos proporcionados

REQUEST

REQUEST BODY - application/json

```
{
  id           integer
  name         string
  members      [integer]
  adminMembers [integer]
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

3.6 DELETE /deleteGroup/{groupId}

Elimina el grupo con el id proporcionado y devuelve un mensaje de error en caso de no ser posible.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*groupId	integer	Id del grupo

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

3.7 POST /changeGroupStatusById

Activa o desactiva al grupo con el id proporcionado, según su estado actual (active o inactive)

REQUEST

REQUEST BODY - application/json

```
{
  groupId  integer
  status   enum     ALLOWED:active, inactive
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

---

4. PROYECTOS Y PLANES DE PRUEBAS

4.1 POST /searchProjects

Devuelve una lista de proyectos que cumplen con los criterios proporcionados

REQUEST

REQUEST BODY - application/json

```
{
  projectName    string
  projectGroups  [integer]
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

4.2 POST /createProject

Crea un proyecto con el nombre proporcionado, perteneciente al grupo indicado

REQUEST

REQUEST BODY - application/json

```
{
  projectName   string
  projectGroup  integer
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

4.3 GET /getProjectById/{projectId}

Devuelve la información del proyecto con el id proporcionado y una lista de sus planes de pruebas

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*projectId	string	Clave compuesta del proyecto, representada por una cadena con el formato `IdGrupo.IdProyecto`

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

4.4 POST /updateProject

Actualiza la información del proyecto con el id proporcionado

El id es la clave compuesta del proyecto, representada por una cadena con el formato IdGrupo.IdProyecto

REQUEST

REQUEST BODY - application/json

```
{
  id          string
  projectName string
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

4.5 DELETE /deleteProject/{projectId}

Elimina el proyecto con el id proporcionado y devuelve un mensaje de error en caso de no ser posible.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*projectId	string	Clave compuesta del proyecto, representada por una cadena con el formato `IdGrupo.IdProyecto`

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

4.6 POST /createTestplan

Crea un plan de pruebas para el proyecto indicado

El id del proyecto (*idproject*) se representa con una cadena en el formato IdGrupo.IdProyecto

REQUEST

REQUEST BODY - application/json

```
{
  idproject    string
  name         string
  description  string
  tags         [string]
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

4.7 POST /searchTestplans

Devuelve una lista de planes de pruebas que cumplen con los criterios proporcionados

group: id de grupo
projects: ids de los proyectos pertenecientes al grupo indicado
createdOn: rango de fechas

REQUEST

REQUEST BODY - application/json

```
{
  group         integer
  projects      [integer]
  testplanName  string
  createdOn     [string]
  tags          [string]
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

4.8 GET /getTestplanById/{testplanId}

Devuelve la información del plan de pruebas con el id proporcionado y una lista de sus casos de pruebas.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*testplanId	string	Clave compuesta del plan de pruebas, representada por una cadena con el formato `IdGrupo.IdProyecto.IdPlan`

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

4.9 POST /updateTestplan

Actualiza la información del plan de pruebas con el id proporcionado

El id es la clave compuesta del plan de pruebas, representada por una cadena con el formato IdGrupo.IdProyecto.IdPlan

REQUEST

REQUEST BODY - application/json

```
{
  id           string
  name         string
  description  string
  tags         [string]
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

4.10 DELETE /deleteTestplan/{testplanId}

Elimina el plan de pruebas con el id proporcionado y devuelve un mensaje de error en caso de no ser posible.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*testplanId	string	Clave compuesta del plan de pruebas, representada por una cadena con el formato `IdGrupo.IdProyecto.IdPlan`

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

4.11 GET /downloadTestplanFile/{filename}

Descarga el archivo .xlsx identificado con el nombre proporcionado.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION

NAME	TYPE	DESCRIPTION
*filename	string	

RESPONSE

STATUS CODE - 200: Un archivo .xlsx

RESPONSE MODEL - application/vnd.openxmlformats-officedocument.spreadsheetml.sheet

```
string
```

STATUS CODE - 404: Entrada inválida

5. REPORTES

5.1 POST /getTestplansTestcasesCount

Devuelve los datos del reporte de número de casos de prueba.

projects: arreglo de cadenas con el formato IdGrupo.IdProyecto

dates: rango de fechas para utilizar como inicio y fin del intervalo del reporte

REQUEST

REQUEST BODY - application/json

```
{
  projects [string]
  dates    [string]
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

5.2 POST /getExecutionsReport

Devuelve los datos del reporte de número de ejecuciones.

projects: arreglo de cadenas con el formato IdGrupo.IdProyecto

dates: rango de fechas para utilizar como inicio y fin del intervalo del reporte

REQUEST

REQUEST BODY - application/json

```
{
  projects [string]
  dates    [string]
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

5.3 GET /getCurrentGroupStats/{groupId}

Devuelve la cuenta de planes de pruebas creados y casos de prueba exportados para el grupo con el id proporcionado.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*groupId	integer	Clave del grupo, `IdGrupo`

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

6. SESIÓN Y USUARIOS

6.1 POST /userLogIn

Autentifica y autoriza al usuario para iniciar sesión

REQUEST

REQUEST BODY - application/json

```
{
  username string
  password string
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

6.2 POST /changePassword

Cambia la contraseña del usuario

REQUEST

REQUEST BODY - application/json

```
{
  userId          string
  currentPassword string
  newPassword     string
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

6.3 GET /getUserInfoById/{userId}

Devuelve la información del usuario según el id proporcionado

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*userId	integer	Id del usuario

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

6.4 POST /createUser

Crea un usuario y envía un mail de verificación

REQUEST

REQUEST BODY - application/json

```
{
  email         string
  username      string
  password      string
  name          string
  lastname      string
  dni           string
  street        string
  streetNumber  string
  addressExtra  string
  job           string
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

6.5 POST /searchUsers

Devuelve una lista de usuarios que cumplen con los criterio provistos

REQUEST

REQUEST BODY - application/json

```
{
  createdOn [string]
  email     string
  name      string
  status    string
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

6.6 POST /updateUser

Actualiza la información del usuario con los datos proporcionados

REQUEST

REQUEST BODY - application/json

```
{
  id            integer
  email         string
  username      string
  name          string
  lastname      string
  dni           string
  street        string
  streetNumber  string
  addressExtra  string
  job           string
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

6.7 DELETE /deleteUserById/{userId}

Elimina el usuario con el id proporcionado y devuelve un mensaje de error en caso de no ser posible.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*userId	integer	Id del usuario

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida

6.8 POST /changeUserStatusById

Activa o desactiva al usuario con el id proporcionado, según su estado actual (active o inactive)

REQUEST

REQUEST BODY - application/json

```
{
  userId  integer
  email   string
  status  enum     ALLOWED:active, inactive
}
```

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 400: Entrada inválida