

SociosResource

Los métodos de esta clase producen un MediaType.APPLICATION_JSON, por lo que los responses tendrán el correspondiente header application/json, su url es <http://localhost:8081/socios>

GET requestSocios()

Este método no consume nada y devuelve una lista en Json de todos los socios que estén guardados en la base de datos. Su url es <http://localhost:8081/socios>. Devuelve el código 200 OK.

Ejemplo:{

```
  "codigo": "XB",
  "email": "ana@asdf-com",
  "nombre": "ANA",
  "saldo": 23.0
},
{
  "codigo": "XC",
  "email": "pepe@asdf-com",
  "nombre": "PEPE",
  "saldo": 8.0
},
{
  "codigo": "1",
  "email": "miguel@upsa.es",
  "nombre": "Miguel",
  "saldo": 32.0
},
{
  "codigo": "XA",
  "email": "enrique@asdf-com",
  "nombre": "ENRIQUE",
  "saldo": 320.0
}
}
```

GET requestSocio(PathParam("codigo"))

Su url es `http://localhost:8081/socios/{codigo}`. Este método devuelve el socio que se ha introducido como PathParam y devuelve la información del socio con código 200 OK. Por ejemplo si introducimos <http://localhost:8081/socios/XA> saldrá el siguiente JSON:

```
{
  "codigo": "XA",
  "email": "enrique@asdf-com",
  "nombre": "ENRIQUE",
  "saldo": 320.0
}
```

GET selectSocios(QueryParam("codigos"))

Un ejemplo de su url es `http://localhost:8081/socios?codigos=XB,XA,XC`. Este método devuelve un JSONArray con los socios que tienen los códigos que están en la lista introducida por el queryParams introducido en la url. El response devuelve el código 200 OK y el siguiente JSONArray con el ejemplo de antes:

```
[
  {
    "codigo": "XB",
    "email": "ana@asdf-com",
    "nombre": "ANA",
    "saldo": 23.0
  },
  {
    "codigo": "XA",
    "email": "enrique@asdf-com",
    "nombre": "ENRIQUE",
    "saldo": 320.0
  },
  {
    "codigo": "XC",
    "email": "pepe@asdf-com",
    "nombre": "PEPE",
    "saldo": 8.0
  }
]
```

POST addSocio(UnidentifiedSocio)

Este método consume un MediaType.APPLICATION_JSON con un UnidentifiedSocio, por ejemplo: {

```
"nombre": "Roberto",
```

```
"email": "roberto@upsa.es"
}
```

La url es <http://localhost:8081/socios> y devuelve un Response con código 200 OK y un entity con un JSON con la información del socio creado, con el código que se le asigne y el saldo por defecto es 0 al crear un nuevo socio, un ejemplo es: {

```
"codigo": "26",
"email": "roberto@upsa.es",
"nombre": "Roberto",
"saldo": 0.0
}
```

PUT updateSocio(Socio)

Este método consume un MediaType.APPLICATION_JSON con Socio, por ejemplo:

```
{
  "codigo": "26",
  "email": "roberto@upsa.es",
  "nombre": "Roberto",
  "saldo": 100.0
}
```

La url es <http://localhost:8081/socios> y devuelve un Response con código 200 OK y un entity con un JSON con la información del socio actualizado(si no hay un socio con ese código se lanza el código 404 Not found), un ejemplo de respuesta es:

```
{
  "código": "26",
  "email": "roberto@upsa.es",
  "nombre": "Roberto",
  "saldo": 100.0
}
```

DELETE

requestDeleteSocio(PathParam("codigo"))

La url es <http://localhost:8081/socios/{codigo}>, este método recibe el path param código para saber que socio tiene que ser borrado ,devuelva un Response con el código 200 OK si existe el socio y lo ha borrado, si no devuelve el código 404 Not Found.