

Universidad ORT Uruguay
Facultad de Ingeniería

Obligatorio 1, Diseño de Aplicaciones 2
Evidencia de las Pruebas con Postman

Entregado como requisito para la obtención del título de Licenciado/a en Sistemas

Stella Steinfeld (229398)
Eugenia Matto (227056)
Santiago Vairo(244023)

2024

<https://github.com/IngSoft-DA2/Steinfeld-Vairo-Matto>

Nosotros, Stella Steinfeld (229398), Eugenia Matto (227056), Santiago Vairo (244023), declaramos que el trabajo que se presenta en esa obra es de nuestra propia mano. Podemos asegurar que:

- La obra fue producida en su totalidad mientras realizábamos diseño de aplicaciones 2;
- Cuando hemos consultado el trabajo publicado por otros, lo hemos atribuido con claridad;
- Cuando hemos citado obras de otros, hemos indicado las fuentes. Con excepción de estas citas, la obra es enteramente nuestra;
- En la obra, hemos acusado recibo de las ayudas recibidas;
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, hemos explicado claramente qué fue contribuido por otros, y qué fue contribuido por nosotros;
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

Abstract	4
Un administrador puede realizar lo siguiente	5
Manejo de notificaciones	8
Disponible para todos los usuarios autenticados (Filtrar por user_id y seen)	8
Un dueño de empresa puede realizar lo siguiente	9
Un dueño de hogar puede realizar lo siguiente	13
Una persona no autenticada puede realizar lo siguiente	16
Una persona autenticada puede realizar lo siguiente	17

Abstract

Este documento detalla las pruebas realizadas con Postman para la aplicación **SmartHub**, un sistema que permite la gestión de dispositivos inteligentes en entornos residenciales y empresariales. **SmartHub** ofrece diversas funcionalidades orientadas a propietarios de hogares, empresas y administradores, facilitando el control centralizado y eficiente de dispositivos conectados.

La aplicación ha sido diseñada para optimizar la administración de los distintos actores involucrados, permitiendo a los usuarios gestionar hogares, configurar dispositivos, establecer permisos de uso y recibir notificaciones relevantes, entre otras acciones clave. Las pruebas descritas en este documento cubren los aspectos funcionales de la API de SmartHub, evaluando el correcto funcionamiento de los endpoints y garantizando la integridad de los datos y la interacción entre los componentes del sistema.

Este trabajo es parte del curso **Diseño de Aplicaciones 2** de la **Universidad ORT Uruguay** y tiene como objetivo proporcionar una referencia clara y estructurada de las pruebas realizadas, explicando las configuraciones de cada prueba, las solicitudes enviadas y las respuestas esperadas de cada funcionalidad del sistema.

Video demostrando dichas pruebas: <https://youtu.be/rHIV49dukNc>

Variables de entorno utilizadas:

	Variable	Type	Initial value
<input checked="" type="checkbox"/>	Token	secret
<input checked="" type="checkbox"/>	Homeld	default	Homeld
<input checked="" type="checkbox"/>	Email	default	Email
<input checked="" type="checkbox"/>	AdminAccountld	default	AdminAccountld
<input checked="" type="checkbox"/>	UserId	default	UserId

Colección de Postman:

<https://elements.getpostman.com/redirect?entityId=23199255-b84d7196-a5f6-43d1-ac62-5889adaa2329&entityType=collection>

Repositorio de Github:

<https://github.com/IngSoft-DA2/Steinfeld-Vairo-Matto>

Requerimientos funcionales

La mayoría de las request requieren token de autorización, se especificará en aquellos que no sea necesario.

<input checked="" type="checkbox"/>	Authorization	{{Token}}
-------------------------------------	---------------	-----------

Validaciones generales:

Para acceder a cualquier endpoint se necesitará el permiso correspondiente, que le será dado al usuario según su rol, por lo cual si quiere acceder y no tiene permiso la respuesta será: (variando el permiso necesario)

Body Cookies Headers (4) Test Results 403 Forbidden · 30 ms · 244 B · Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "innerCode": "Forbidden",
3   "message": "Missing permission createadminaccount-administrator"
4 }
```

Un administrador puede realizar lo siguiente

Mantenimiento de cuentas de administrador

Crear cuenta de administrador:

SmartHub / Administrator / AddAdminAccount Save Share

POST {{SmartHub_Host}}/api/administrators Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "Name": "Daniel",
3   "Surname": "Acevedo",
4   "Email": "dani@gmail.com",
5   "Password": "danielacevedo123"
6 }
```

Body Cookies Headers (4) Test Results 200 OK · 875 ms · 196 B · Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "token": "137166c1-443c-430a-713d-08dce72174e9"
3 }
```

Si se intenta agregar un administrador con email existente (validación aplicada en la creación de los 3 tipos de usuarios):

Body Cookies Headers (4) Test Results 500 Internal Server Error · 123 ms · 305 B · Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "innerCode": "InternalServerError",
3   "message": "There was an error when processing the request: User with email: juana@gmail.com already exists."
4 }
```

Validación de campos (Aplicada en la creación de los 3 tipos de usuarios):

Body Cookies Headers (4) Test Results 400 Bad Request 57 ms 543 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "innerCode": "ValidationError",
3   "message": "El request es inválido",
4   "errors": {
5     "Name": [
6       "Name is required",
7       "Name cannot be empty"
8     ],
9     "Email": [
10      "The email must be in a valid format (e.g., user@domain.com).",
11    ],
12    "Surname": [
13      "Surname is required",
14      "Surname cannot be empty"
15    ],
16    "Password": [
17      "The password must be at least 6 characters long.",
18      "The password must contain at least one special character."
19    ]
20  }
21 }
```

Activar Windows
Ve a Configuración para

Si no se mandan datos en body:

SmartHub / Administrator / AddAdminAccount Save Share

POST {{SmartHub_Host}}/api/administrators Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2 }
```

Body Cookies Headers (4) Test Results 400 Bad Request 681 ms 438 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "innerCode": "ValidationError",
3   "message": "El request es inválido",
4   "errors": {
5     "$": [
6       "JSON deserialization for type 'Model.AdministratorRequest' was missing required properties, including the following: name; surname; email; password"
7     ]
8   }
9 }
```

Borrar cuenta de administrador:

SmartHub / Administrator / DeleteAdminAccount Save Share

DELETE {{SmartHub_Host}}/api/administrators/{{AdminAccountId}} Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Headers 6 hidden

	Key	Value	Description		Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	{{Token}}				

Body Cookies Headers (4) Test Results 200 OK 789 ms 221 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Deleted Admin account: 56c9f2b8-7d76-4f4c-78e0-08dce72174a2"
3 }
```

Si intentamos borrar cuenta que no es Administrador o Id inválido:

Body Cookies Headers (4) Test Results 500 Internal Server Error 87 ms 280 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "innerCode": "InternalServerError",
3   "message": "There was an error when processing the request: Administrator not found"
4 }
```

Creación de cuentas para dueños de empresas de dispositivos inteligentes:

SmartHub / CompanyOwner / AddCompanyOwner

SaveShare

POST{{SmartHub_Host}}/api/company-ownersSend

ParamsAuthorizationHeaders (9)BodyScriptsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

Beautify

```
1 {
2   "Name": "Eugenia",
3   "Surname": "Matto",
4   "Email": "euge@gmail.com",
5   "Password": "ho!!lla"
6 }
```

BodyCookiesHeaders (4)Test Results

200 OK160 ms196 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "token": "f505ac7a-d97f-405c-08b1-08dce7248c51"
3 }
```

Listar todas las cuentas: Disponible con paginación pedida.

SmartHub / Account / GetAccounts

SaveShare

GET{{SmartHub_Host}}/api/accounts?role=HomeOwner&fullName=Novak DjokovicSend

ParamsAuthorizationHeaders (7)BodyScriptsSettings

BodyCookiesHeaders (4)Test Results

200 OK46 ms329 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "items": [
3     {
4       "name": "Novak",
5       "surname": "Djokovic",
6       "fullName": "Novak Djokovic",
7       "role": "HomeOwner",
8       "createdAt": "2024-10-07T20:19:52.283317"
9     }
10  ],
11   "totalCount": 1,
12   "pageNumber": 1,
13   "pageSize": 10
14 }
```

SmartHub / Account / GetAccounts

SaveShare

GET{{SmartHub_Host}}/api/accounts?role=CompanyOwner&pageNumber=2&pageSize=3Send

ParamsAuthorizationHeaders (7)BodyScriptsSettings

BodyCookiesHeaders (4)Test Results

200 OK20 ms590 BSave Response

PrettyRawPreviewVisualizeJSON

```
1 {
2   "items": [
3     {
4       "name": "Pete",
5       "surname": "Sampxas",
6       "fullName": "Pete Sampxas",
7       "role": "CompanyOwner",
8       "createdAt": "2024-10-07T20:22:24.7589654"
9     },
10    {
11      "name": "Alexander",
12      "surname": "Zverev",
13      "fullName": "Alexander Zverev",
14      "role": "CompanyOwner",
15      "createdAt": "2024-10-07T20:22:58.6778521"
16    },
17    {
18      "name": "Stan",
19      "surname": "Wawrinka",
20      "fullName": "Stan Wawrinka",
21      "role": "CompanyOwner",
22      "createdAt": "2024-10-07T20:22:48.2681911"
23    }
24  ],
25   "totalCount": 9,
26   "pageNumber": 2,
27   "pageSize": 3
28 }
```

Listar todas las empresas: Disponible con paginación pedida.

SmartHub / Company / GetCompanies

GET `{{SmartHub_Host}}/api/companies?companyName=SA&pageNumber=2&pageSize=2` Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Headers 6 hidden

Key	Value	Description	Bulk Edit	Presets
Authorization	{{Token}}			

Body Cookies Headers (4) Test Results 200 OK 30 ms 466 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "items": [
3     {
4       "name": "CatMania SA",
5       "logo": "https://car.com/logo.png",
6       "rut": "7778212",
7       "owner": "Stan Mawrinka",
8       "ownerEmail": "swawrinka@gmail.com"
9     },
10    {
11      "name": "Tench SA",
12      "logo": "https://tench.com/logo.png",
13      "rut": "6863422",
14      "owner": "Alexander Zverev",
15      "ownerEmail": "azverev@gmail.com"
16    }
17  ],
18  "totalCount": 4,
19  "pageNumber": 2,
20  "pageSize": 2
21 }
```

SmartHub / Company / GetCompanies

GET `{{SmartHub_Host}}/api/companies?fullName=Kim Clijsters` Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Headers 6 hidden

Key	Value	Description	Bulk Edit	Presets
Authorization	{{Token}}			

Body Cookies Headers (4) Test Results 200 OK 24 ms 337 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "items": [
3     {
4       "name": "Salta Inc",
5       "logo": "https://saltis.com/logo.png",
6       "rut": "1188621",
7       "owner": "Kim Clijsters",
8       "ownerEmail": "kclijsters@gmail.com"
9     }
10  ],
11  "totalCount": 1,
12  "pageNumber": 1,
13  "pageSize": 10
14 }
```

Manejo de notificaciones

Disponible para todos los usuarios autenticados (Filtra por user_id y seen)

SmartHub / Notification / GetNotifications

GET `{{SmartHub_Host}}/api/notifications?user_id=8C83A0FA-0FA3-444A-838E-08DCE7267241&seen=false` Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Body Cookies Headers (4) Test Results 200 OK 64 ms 1.04 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": "60c5dbe1-f21e-4a74-913a-14b29e83efb0",
4     "event": "Camera Alert: Motion detected",
5     "seen": false,
6     "email": "ndjokovic@gmail.com"
7   },
8   {
9     "id": "8e1ffa4c-9ee3-4092-b0fc-34503f4e7e23",
10    "event": "Camera Alert: Person detected: Juan",
11    "seen": false,
12    "email": "ndjokovic@gmail.com"
13  },
14  {
15    "id": "c4004690-01eb-426b-9d92-4782d489340d",
16    "event": "Window Sensor Alert: Open",
17    "seen": false,
18    "email": "ndjokovic@gmail.com"
19  },
20  {
21    "id": "fd938a9c-31b2-4fca-a3ab-609a4b014145",
22    "event": "Camera Alert: Person detected: dani@gmail.com",
23    "seen": false
24  }
25 ]
```


Un dueño de empresa puede realizar lo siguiente

Creación de una empresa (Con validaciones para todos sus campos)

SmartHub / Company / AddCompany

POST {{SmartHub_Host}} /api/companies

Send

Params Auth Headers (9) Body Scripts Settings

raw JSON Beautify

```
1 {
2   "name": "CatMania SA",
3   "logoUrl": "https://car.com/logo.png",
4   "rut": "7778212"
5 }
```

Body 200 OK • 104 ms • 193 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "c9007ed8-e773-4c4d-9ac6-3a552c776f2f"
3 }
```

SmartHub / Company / AddCompany

POST {{SmartHub_Host}} /api/companies

Send

Params Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Beautify

```
1 {
2   "name": "Julimania SA",
3   "logoUrl": "https://jujul.com/logo.png",
4   "rut": "9870003"
5 }
```

Body Cookies Headers (4) Test Results 500 Internal Server Error • 233 ms • 287 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "innerCode": "InternalServerError",
3   "message": "There was an error when processing the request: Already the owner of a company"
4 }
```

SmartHub / Company / AddCompany

POST {{SmartHub_Host}} /api/companies

Send

Params Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Beautify

```
1 {
2 }
```

Body Cookies Headers (4) Test Results 400 Bad Request • 58 ms • 418 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "innerCode": "ValidationError",
3   "message": "El request es inválido",
4   "errors": {
5     "$": [
6       "JSON deserialization for type 'Model.CompanyRequest' was missing required properties, including the following: name; logoUrl; rut"
7     ]
8   }
9 }
```

Pretty Raw Preview Visualize JSON

```
1 {
2   "innerCode": "ValidationError",
3   "message": "El request es inválido",
4   "errors": {
5     "Rut": [
6       "RUT can only contain numbers"
7     ]
8   }
9 }
```

Validaciones con campos vacíos:

SmartHub / Company / AddCompany

POST {{SmartHub_Host}}/api/companies

Params Authorization Headers (9) Body Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☒ JSON

```
1 {
2   "name": "",
3   "logoUrl": "",
4   "rut": ""
5 }
```

Body Cookies Headers (4) Test Results 400 Bad R

Pretty Raw Preview Visualize JSON

```
1 {
2   "innerCode": "ValidationError",
3   "message": "El request es inválido",
4   "errors": {
5     "Rut": [
6       "RUT is required",
7       "RUT cannot be empty"
8     ],
9     "Name": [
10      "Name is required",
11      "Name cannot be empty"
12    ],
13    "LogoUrl": [
14      "Logo is required",
15      "Logo cannot be empty"
16    ]
17  }
18 }
```

Registrar cámaras de seguridad (Con validaciones para todos sus campos)

SmartHub / SecurityCam / AddSecurityCam Save Share

POST {{SmartHub_Host}}/api/cameras Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☒ JSON

```
1 {
2   "Name": "Camera Name",
3   "ModelNumber": "ABC123",
4   "Description": "Description of the camera",
5   "Photos": ["url1.jpg", "url2.jpg"],
6   "UseTypes": [
7     "Outdoor",
8     "Indoor"
9   ],
10  "SupportedActions": [
11    "MotionDetection",
12    "PersonDetection"
13  ]
14 }
```

Body Cookies Headers (4) Test Results 200 OK 118 ms 193 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": "1784145a-4c95-409e-adfd-4cf0b2819f2c"
3 }
```

Registrar sensor de ventanas (Con validaciones para todos sus campos)

The screenshot shows a REST client interface with the following details:

- URL:** `{{SmartHub_Host}}/api/sensors`
- Method:** POST
- Body (JSON):**

```
{
  "Name": "Temperature Sensor 4",
  "ModelNumber": "TS-131",
  "Description": "A high-precision temperature sensor for home automation.",
  "Photos": [
    "https://example.com/photo1.jpg",
    "https://example.com/photo2.jpg"
  ]
}
```
- Response:** 200 OK, 58 ms, 193 B. The response body is:

```
{
  "id": "28c1c90d-6ac0-4f12-8962-bdd49b7d935f"
}
```

Entendemos por diseño que cualquier usuario autenticado puede enviar una notificación, lo correcto sería realizar autenticación al dispositivo pero no estaba en el alcance de esta entrega.

Cámara: Detección de persona (se le pasa usuario opcional por parámetro)

The screenshot shows a REST client interface with the following details:

- URL:** `{{SmartHub_Host}}/api/cameras/person-detection?user_detection=dani@gmail.com&hardware_id=8644c635-be37-4db2-f736-08dce737900b`
- Method:** POST
- Response:** 200 OK, 52 ms, 595 B. The response body is:

```
{
  "notifications": [
    {
      "id": "fd938a9c-31b2-4fca-a3ab-689a4b814145",
      "event": "Camera Alert: Person detected: dani@gmail.com",
      "seen": false,
      "email": "ndjokovic@gmail.com"
    },
    {
      "id": "962b39f7-12bf-4492-a074-9073cecf7876",
      "event": "Camera Alert: Person detected: dani@gmail.com",
      "seen": false,
      "email": "sgraf@gmail.com"
    },
    {
      "id": "b7eb33c0-a432-4087-abec-717301b18a76",
      "event": "Camera Alert: Person detected: dani@gmail.com",
      "seen": false,
      "email": "rlavex@gmail.com"
    }
  ]
}
```

Cámara: Detección de movimiento

The screenshot shows a REST client interface with the following details:

- URL:** `{{SmartHub_Host}}/api/cameras/motion-detection?hardware_id=8644c635-be37-4db2-f736-08dce737900b`
- Method:** POST
- Response:** 200 OK, 60 ms, 547 B. The response body is:

```
{
  "notifications": [
    {
      "id": "60c5dbe1-f21e-4a74-913a-14b29e83efb0",
      "event": "Camera Alert: Motion detected",
      "seen": false,
      "email": "ndjokovic@gmail.com"
    },
    {
      "id": "6d5da358-e0ab-4c67-b2aa-d8a848fba537",
      "event": "Camera Alert: Motion detected",
      "seen": false,
      "email": "sgraf@gmail.com"
    }
  ]
}
```

Validación dispositivo no es cámara de seguridad:

SmartHub / SecurityCam / SecurityCam MotionDetection

POST `{{SmartHub_Host}}/api/cameras/motion-detection?hardware_id=7d31b75b-a152-49fe-de48-08dce7b0e3e1` Send

Params Authorization Headers (8) Body Scripts Settings

Body Cookies Headers (4) Test Results 500 Internal Server Error 62 ms 338 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "innerCode": "InternalServerError",
3   "message": "There was an error when processing the request: Hardware 7d31b75b-a152-49fe-de48-08dce7b0e3e1 does not have a SecurityCam device."
4 }
```

Validación cámara no soporta esa acción (disponible en las dos notificaciones):

SmartHub / SecurityCam / SecurityCam MotionDetection

POST `{{SmartHub_Host}}/api/cameras/motion-detection?hardware_id=f2772bd2-2233-4d1f-de4a-08dce7b0e3e1` Send

Params Authorization Headers (8) Body Scripts Settings

Body Cookies Headers (4) Test Results 500 Internal Server Error 48 ms 340 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "innerCode": "InternalServerError",
3   "message": "There was an error when processing the request: SecurityCam f2772bd2-2233-4d1f-de4a-08dce7b0e3e1 does not support Motion Detection."
4 }
```

Sensor: Close / Open

SmartHub / WindowSensor / WindowSensor CloseNotification

POST `{{SmartHub_Host}}/api/sensors/open?hardware_id=de656366-cb05-401a-f735-08dce737900b` Send

Params Authorization Headers (8) Body Scripts Settings

Query Params

<input checked="" type="checkbox"/> Key	Value	Description	Bulk Edit
---	-------	-------------	-----------

Body Cookies Headers (4) Test Results 200 OK 66 ms 535 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "notifications": [
3     {
4       "id": "ca5c8c58-771f-4f94-952d-990e63b84a4c",
5       "event": "Window Sensor Alert: Open",
6       "seen": false,
7       "email": "ndjokovic@gmail.com"
8     },
9     {
10      "id": "ae13e89b-660a-48a4-be77-1e04ef37e3c0",
11      "event": "Window Sensor Alert: Open".
12    }
13  ]
14 }
```

SmartHub / WindowSensor / WindowSensor OpenNotification

POST `{{SmartHub_Host}}/api/sensors/open?hardware_id=de656366-cb05-401a-f735-08dce737900b` Send

Params Authorization Headers (8) Body Scripts Settings

Headers 7 hidden

Key	Value	Description	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	<code>{{Token}}</code>			

Body Cookies Headers (4) Test Results 200 OK 93 ms 535 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "notifications": [
3     {
4       "id": "c4004690-01eb-426b-9d92-4782d489340d",
5       "event": "Window Sensor Alert: Open",
6       "seen": false,
7       "email": "ndjokovic@gmail.com"
8     },
9     {
10      "id": "b4a75ac9-22bb-4170-9ca5-9d3521014df4",
11      "event": "Window Sensor Alert: Open".
12    }
13  ]
14 }
```

Un dueño de hogar puede realizar lo siguiente

Creación de hogares (Con validaciones para todos sus campos)

SmartHub / Home / AddHome

POST {{SmartHub_Host}}/api/homes

Params Authorization Headers (9) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "Street": "Jackson",
3   "StreetNumber": "877",
4   "Ubication": {
5     "Lat": 66,
6     "Lon": -69
7   },
8   "MaxMembers": 3
9 }
```

Body Cookies Headers (4) Test Results 200 OK • 76 ms • 198 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "home_id": "15ff787f-ada2-4f3e-7a9a-08dce72d8124"
3 }
```

Agregar miembros al hogar

SmartHub / Member / AddHomeMember

POST {{SmartHub_Host}}/api/members?homeId={{HomeId}}

Params Authorization Headers (9) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "memberEmail": "rlaver@gmail.com",
3   "Permissions": [
4     "AssociateDevices",
5     "ListDevices"
6   ]
7 }
```

Body Cookies Headers (4) Test Results 200 OK • 853 ms • 188 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Member added successfully."
3 }
```

Body Cookies Headers (4) Test Results 500 Internal Server Error • 3.96 s • 304 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "innerCode": "InternalServerError",
3   "message": "There was an error when processing the request: No user found with email: cwozniacksi@gmail.com"
4 }
```

Body Cookies Headers (4) Test Results 500 Internal Server Error • 194 ms • 327 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "innerCode": "InternalServerError",
3   "message": "There was an error when processing the request: User with Id: mnnavratilova@gmail.com is already a member of this home."
4 }
```


Listado de dispositivos en un hogar:

SmartHub / Home / GetHomeDevice

GET{{SmartHub_Host}}/api/homes/{{HomelId}}/devices

ParamsAuthorizationHeaders (7)BodyScriptsSettings

Headers6 hidden

Key	Value	Description

BodyCookiesHeaders (4)Test Results200 OK33 ms1.13 KB

PrettyRawPreviewVisualizeJSON

```
1 [
2   {
3     "id": "de656366-cb05-401a-f735-08dce737900b",
4     "online": true,
5     "deviceInfo": {
6       "id": "edfde629-1c5a-4e11-ba35-c2e21cd6970",
7       "name": "Window Sensor megaS",
8       "model": "KS-133",
9       "mainPhoto": "https://megaK.com/photo1.jpg",
10      "photos": [
11        "https://megaK.com/photo1.jpg",
12        "https://megaK.com/photo2.jpg"
13      ],
14      "type": "WindowSensor",
15      "description": "A rapid window sensor for home automation.",
16      "companyName": "Universidad ORT"
17    }
18  },
19  {
20    "id": "ccfb4375-181e-4d09-f737-08dce737900b",
21    "online": true,
22    "deviceInfo": {
23      "id": "1784145a-4c95-409e-adfd-4cf0b2819f2c",
24      "name": "Camera Name",
25      "model": "ABC123",
26      "mainPhoto": "url1.jpg",
27      "photos": [
28        "url1.jpg",
29        "url2.jpg"
30      ],
31      "type": "SecurityCam",
32      "description": "Description of the camera",
33      "companyName": "Tench SA"
34    }
35  }
36 ]
```

Configuración de notificaciones para miembros:

SmartHub / Member / AddHomeNotificationMembers

PATCH{{SmartHub_Host}}/api/members/{{HomelId}}/notification

ParamsAuthorizationHeaders (9)BodyScriptsSettings

noneform-datax-www-form-urlencoderauwbinaryGraphQLJSON

1 [
2 "ndjokovic@gmail.com", "sgraf@gmail.com", "rlaver@gmail.com"
3]

BodyCookiesHeaders (4)Test Results200 OK104 ms202 B

PrettyRawPreviewVisualizeJSON

```
1 {
2   "message": "Notification members added successfully."
3 }
```

Valida si el usuario es parte del hogar:

BodyCookiesHeaders (4)Test Results500 Internal Server Error5.88 s

PrettyRawPreviewVisualizeJSON

```
1 {
2   "innerCode": "InternalServerError",
3   "message": "There was an error when processing the request: No user ndjokovic@gmail.com found."
4 }
```

Una persona no autenticada puede realizar lo siguiente

Autenticación mediante credenciales (No requiere token)

SmartHub / Login

POST {{SmartHub_Host}}/api/user/login

Params Authorization Headers (8) Body Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "Email": "vwilliams@gmail.com",
3   "Password": "Vwil@0"
4 }
```

Body Cookies Headers (4) Test Results 200 OK

Pretty Raw Preview Visualize JSON

```
1 {
2   "token": "2344748e-3e8f-4a7e-53c7-08dce7369309"
3 }
```

SmartHub / Login

POST {{SmartHub_Host}}/api/user/login

Params Authorization Headers (8) Body Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "Email": "vwilliams@gmail.com",
3   "Password": "Vwil@0"
4 }
```

Body Cookies Headers (4) Test Results 400 Bad Request • 108 ms • 242 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "innerCode": "ValidationError",
3   "message": "Invalid username or password."
}
```

Creación de cuenta para dueño de hogar (No requiere token)

SmartHub / HomeOwner / AddHomeOwner

POST {{SmartHub_Host}}/api/home-owners

Params Authorization Headers (9) Body Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "Name": "Venus",
3   "Surname": "Williams",
4   "Email": "vwilliams@gmail.com",
5   "Password": "Vwil@0",
6   "ProfilePhoto": "vwilliams.jpg"
7 }
```

Body Cookies Headers (4) Test Results 200 OK

Pretty Raw Preview Visualize JSON

```
1 {
2   "token": "0867f28d-6d9c-4dd5-08ba-08dce7248c51"
3 }
```


Una persona autenticada puede realizar lo siguiente

Listado de dispositivos (disponible con paginación)

HTTP SmartHub / Device / GetDevices

GET `{{SmartHub_Host}}/api/devices?name=Camera&model=AS&pageNumber=0&pageSize=2`

Params • Authorization Headers (7) Body Scripts Settings

<input checked="" type="checkbox"/>	Authorization	<code>{{Token}}</code>	
	Key	Value	Description

Body Cookies Headers (4) Test Results **200 OK** • 23 ms •

Pretty Raw Preview Visualize JSON

```
1 {
2   "items": [
3     {
4       "id": "7fa0f5a9-0d22-4416-9d6e-c86eb3187b51",
5       "name": "Camera PRO XX8",
6       "model": "AS-102",
7       "mainPhoto": "url221.jpg",
8       "companyName": "Universidad ORT"
9     },
10    {
11      "id": "5be3cc96-3541-4579-b04f-cfa89a1bcf02",
12      "name": "Camera PRO XX7",
13      "model": "AS-101",
14      "mainPhoto": "url221.jpg",
15      "companyName": "Universidad ORT"
16    }
17  ],
18  "totalCount": 2,
19  "pageNumber": 0,
20  "pageSize": 2
21 }
```

Listado de tipos de dispositivos soportados

HTTP SmartHub / DeviceTypes / GetDeviceTypes Save Share

GET `{{SmartHub_Host}}/api/device-types` **Send**

Params • Auth Headers (7) Body Scripts Settings **Cookies**

Headers 6 hidden

	Key	Value	D.	...	Bulk Edit	Presets	...
--	-----	-------	----	-----	-----------	---------	-----

Body **200 OK** • 29 ms • 178 B •

Pretty Raw Preview Visualize JSON

```
1 [
2   "SecurityCam",
3   "WindowSensor"
4 ]
```