

# PLAYMOBIL

## Descripción

---

Nos vamos a encontrar que los Playmobil están organizados en series históricas.

Cada serie histórica ofrece una serie de productos en formato de caja con un precio de venta. La caja está formada por figuras en distintas cantidades.

Finalmente, las cajas están formadas por figuras que no se pueden adquirir de forma individual, siempre formarán parte de una caja.

## Arranque del contenedor

---

### Opcion 1

Es preferible que se inicie todo con Docker-compose.

El comando

```
docker-compose up -d
```

descarga los contenedores y los pone en marcha, mientras que

```
docker-compose down
```

detiene los contenedores y los elimina

### Opción 2

Suponemos que el servidor de autenticación se descargó previamente y se pone en marcha con

```
docker start authentication_container
```

El contenedor se descarga y arranca con las instrucciones:

```
docker pull adomargon/server_playmobil:0.15
```

```
docker run --name playmobil_container --p 8082:8082  
adomargon/server_playmobil:0.15
```

A partir de este momento tenéis en marcha un servidor web.

## Endpoints

---

Los endpoints están programados para devolver una respuesta después de varios segundos. También dan error con cierta frecuencia. Hay que tener en cuenta ambas situaciones.

### Autenticación de usuarios

El usuario se autentica en el siguiente endpoint empleando el método de envío POST.

```
http://127.0.0.1:8081/api/login
```

## Series de Playmobil

Una serie es un grupo de productos relacionados por una temática.

<http://127.0.0.1:8082/api/series>

## Series de Playmobil por uuid

Se suministra en la url el código de la serie y obtenemos el detalle de dicha serie.

<http://127.0.0.1:8082/api/serie/1703135e-03b3-4a32-b8df-16965d19b862>

## Cajas de Playmobil

Una caja es un producto preparado para la venta.

<http://127.0.0.1:8082/api/boxes>

## Cajas de Playmobil por uuid

Se suministra en la url el código de la caja y obtenemos el detalle de dicha caja.

<http://127.0.0.1:8082/api/box/5297d100-cc4a-4893-abd9-b743efab8838>

## Cajas de una determinada serie

Es posible recuperar expresamente las cajas asociadas a una determinada serie. Para ello se concatena en la url el uuid de la serie. La serie debe existir.

<http://127.0.0.1:8082/api/serie/a0766aed-1f58-46e7-b128-5aabf2ceac51/boxes>

## Figuras de Playmobil

Una figura es un producto individual que formará parte de una caja.

<http://127.0.0.1:8082/api/figures>

## Figuras de Playmobil por uuid

Se suministra en la url el código de la figura y obtenemos el detalle de dicha figura.

<http://127.0.0.1:8082/api/figure/104758d6-5cce-4a2e-a66c-c5bf917b0202>

## Figuras de una determinada caja

Es posible recuperar expresamente las figuras asociadas a una determinada caja. Para ello se concatena en la url el uuid de la caja. La caja debe existir.

<http://127.0.0.1:8082/api/box/5297d100-cc4a-4893-abd9-b743efab8838/figures>

## Proyecto

---

### Página de autenticación de usuarios

Desarrolla una página con un formulario para la autenticación de usuarios. Debe tener dos campos para solicitar el nombre de usuario y la contraseña.

Los ficheros se llamarán:

- login.html
- login.js

Emplea el endpoint de validación de usuarios consumiendo el endpoint correspondiente. Si la validación tiene éxito se recibe un token que habrá que emplear para poder consumir información de los endpoints de Playmobil. El token hay que almacenarlo en una variable de sesión y redirigir a la página de selección de series de Playmobil.

## Página de selección de una serie

Desarrolla una página en la que se muestren todas las series. Se mostrarán en una tabla con una única columna que muestra la denominación de la serie.

Los ficheros se llamarán:

- choose\_serie.html
- choose\_serie.js

Primero hay que comprobar que el usuario se ha autenticado consultando si el token está almacenado en una variable de sesión. Si no hay token hay que redirigir al usuario a la página de login. Si hay token hay que extraer del token el nombre y el apellido para mostrarlo en la cabecera.

Emplea atributos inventados (data-) para colocar en cada fila el identificador de la serie. Luego desde JavaScript tendrás que recuperar dicho atributo inventado (dataset) para saber sobre qué serie se ha clicado.

Al clicar en una serie debe ir a la página de selección de una caja. Hay que hacer una redirección con JavaScript.

Durante la carga de la información deben mostrarse mensajes informativos adecuados: espere a que la información esté disponible, error, ...

## Página de selección de una caja

Desarrolla una página en la que se muestren las cajas de una determinada serie.

El identificador de la serie debe ser extraída de la url. La url seguirá el siguiente formato (probado desde Live Server).

```
http://127.0.0.1:5000/views/choose_box.html?serie=1703135e-03b3-4a32-b8df-16965d19b862
```

Los ficheros se llamarán:

- choose\_box.html
- choose\_box.js

Las cajas se muestran en una tabla con dos columnas, la primera con la denominación y la segunda con el precio. Emplea atributos inventados (data-) para colocar en cada etiqueta información relevante.

## Página de figuras de una caja

Desarrolla una página en la que se muestren las figuras de una determinada caja.

El identificador de la caja debe ser extraída de la url. La url seguirá el siguiente formato.

[http://127.0.0.1:5000/views/figures\\_box.html?box=5297d100-cc4a-4893-abd9-b743efab8838](http://127.0.0.1:5000/views/figures_box.html?box=5297d100-cc4a-4893-abd9-b743efab8838)

Los ficheros se llamarán:

- figures\_box.html
- figures\_box.js

Las figuras se muestran en una tabla con cuatro columnas, la primera con la denominación, la segunda con el código de barras, la tercera con la cantidad y la cuarta con la imagen. Emplea atributos inventados (data-) para colocar en cada etiqueta información relevante.

Las imágenes se encuentran en rutas como la siguiente:

<http://127.0.0.1:8082/assets/roman-soldier.jpg>