ajax

Crea un Observable para una petición Ajax

▶ Signatura

Descripción

Crea un Observable para una petición Ajax a partir de un objeto de petición con la url, cabeceras etc. o a partir de una URL.

Ejemplos

Realizar una petición Ajax, y emitir el objeto AjaxResponse completo

StackBlitz

```
import { ajax } from "rxjs/ajax";

const ghibliFilmsResponse$ = ajax("https://ghibliapi.herokuapp.com/films");

ghibliFilmsResponse$.subscribe(console.log);
// Salida: AjaxResponse { ...request: {...}, status: 200...}
```

Emitir únicamente los datos del objeto respuesta

StackBlitz

```
import { ajax } from "rxjs/ajax";
import { mergeAll } from "rxjs/operators";

const ghibliFilm$ = ajax
  .getJSON("https://ghibliapi.herokuapp.com/films")
  .pipe(mergeAll());

ghibliFilm$.subscribe(console.log);
/* Salida:
{ ...title: 'Castle in the Sky'... },
{ ...title: 'Grave of the Fireflies'... },
{ ...title: 'My Neighbor Totoro'... }...
*/
```

Utilizar un objeto de configuración para los parámetros de la petición AJAX

StackBlitz

```
import { ajax } from "rxjs/ajax";

const ghibliFilmWithHeaders$ = ajax({
  url: "https://ghibliapi.herokuapp.com/films",
  method: "GET",
  headers: {
    "Content-Type": "json",
  },
  body: {
    message: "Mensaje personalizado, porque podemos ;)",
  },
});
ghibliFilmWithHeaders$.subscribe(console.log);
// Salida: AjaxResponse {xhr: {}, request: {}...}
```

Realizar varias peticiones Ajax mediante un operador de proyección de orden superior (mergeMap, switchMap, concatMap, exhaustMap)

StackBlitz

```
import { from, of } from "rxjs";
import { ajax } from "rxjs/ajax";
```

```
import { catchError, mergeMap } from "rxjs/operators";

const filmId$ = of(
  "58611129-2dbc-4a81-a72f-77ddfc1b1b49",
  "2baf70d1-42bb-4437-b551-e5fed5a87abe"
);

function getGhibliFilm(id: string) {
  return ajax.getJSON(`https://ghibliapi.herokuapp.com/films/${id}`);
}

filmId$.pipe(mergeMap((id) => getGhibliFilm(id))).subscribe(console.log);
// Salida: {...title: 'Castle in the Sky'...}, {...title: 'My Neighbor Toto
```

Ejemplos de la documentación oficial

Usar ajax() para obtener el objeto de respuesta que retorna la API

```
import { ajax } from "rxjs/ajax";
import { map, catchError } from "rxjs/operators";
import { of } from "rxjs";

const obs$ = ajax(`https://api.github.com/users?per_page=5`).pipe(
  map((userResponse) => console.log("users: ", userResponse)),
  catchError((error) => {
    console.log("error: ", error);
    return of(error);
  })
);
```

Usar ajax.getJSON() para obtener datos de la API

```
import { ajax } from "rxjs/ajax";
import { map, catchError } from "rxjs/operators";
import { of } from "rxjs";

const obs$ = ajax.getJSON(`https://api.github.com/users?per_page=5`).pipe(
  map((userResponse) => console.log("users: ", userResponse)),
  catchError((error) => {
```

```
    console.log("error: ", error);
    return of(error);
  })
);
```

Usar ajax() con un objeto como argumento y el método POST con un retraso de 2 segundos

```
import { ajax } from "rxjs/ajax";
import { of } from "rxjs";

const users = ajax({
  url: "https://httpbin.org/delay/2",
  method: "POST",
  headers: {
    "Content-Type": "application/json",
    "rxjs-custom-header": "Rxjs",
  },
  body: {
    rxjs: "Hello World!",
  },
}).pipe(
  map((response) => console.log("response: ", response)),
  catchError((error) => {
    console.log("error: ", error);
    return of(error);
  })
);
```

Copy

Usar ajax() para hacer una llamada a la API, que devuelve un objeto error