# Exercise: Source Control Systems

# Stage 5, Task 3 – Create and Edit Ads

**To begin, checkout locally the branch Create-Edit-Ads created in Task 5-0 and make sure it’s up-to-date with the master from Stage 3.**

It’s not good business to have only administrators be able to create ads – it places a great burden on our staff. It’s better to provide an interface for users to create and publish their own ads and the ability to edit or delete them when they wish. We’ll do all of that in a separate branch, while the rest of the app is published and operational.

At the end of this stage, your files should be identical to the ones in folder **5. Branch2**

|  |
| --- |
| **Each task should be assigned to a different team member. Work on tasks 5-1, 5-2 and 5-3 simultaneously and commit changes to the Create-Edit-Ads branch in order.** |

We are inserting four new endpoints – user info, create ad, update (edit) ad and delete ad. Add the highlighted lines to the back-end script:

|  |
| --- |
| api.js |
| *↑*  *Existing code here …*  *// Get user info* $.mockjax(**function** (requestSettings) {  **if** (requestSettings.url.match(/https:\/\/mock\.api\.com\/user\/kid\_rk\/(.+)/)) {  **let** userId = requestSettings.url.match(/https:\/\/mock\.api\.com\/user\/kid\_rk\/(.+)/)[1];  **return** {  response: **function** (origSettings) {  **if** (requestSettings.headers[**"Authorization"**].includes(**"Kinvey mock\_token"**)) {  **let** target = users.filter(u => u.\_id === Number(userId));  **this**.responseText = target.shift();  } **else** {  **this**.**status** = 403;  **this**.responseText = **"You are not authorized"**;  }  }  };  }  });   *// Loading of adverts* $.mockjax(**function** (requestSettings) {  **if** (requestSettings.url===**"https://mock.api.com/appdata/kid\_rk/adverts"** &&  requestSettings.method === **"GET"**) {  **return** {  response: **function** (origSettings) {  **if** (requestSettings.headers[**"Authorization"**].includes(**"Kinvey mock\_token"**)) {  **this**.responseText = adverts;  } **else** {  **this**.**status** = 403;  **this**.responseText = **"You are not authorized"**;  }  }  };  }  });   *// Create advert* $.mockjax(**function** (requestSettings) {  **if** (requestSettings.url === **"https://mock.api.com/appdata/kid\_rk/adverts"** &&  requestSettings.method === **"POST"**) {  **return** {  response: **function** (origSettings) {  **if** (requestSettings.headers[**"Authorization"**].includes(**"Kinvey mock\_token"**)) {  **let** data = requestSettings.data;  **let** lastId = 0;  **if** (adverts.**length** > 0) {  lastId = adverts.map(a => a.\_id).sort((a, b) => b - a)[0];  }  **let** token = requestSettings.headers[**"Authorization"**].replace(**"Kinvey "**, **""**);  **let** creator = users.filter(u => u.\_kmd.authtoken === token)[0].\_id;  **let** advert = {  **\_id**: ++lastId,  **\_acl**: {  **creator**: creator  },  **title**: data.title,  **publisher**: data.**publisher**,  **datePublished**: data.**datePublished**,  **price**: data.**price** };  adverts.push(advert);  **this**.responseText = advert;  } **else** {  **this**.**status** = 403;  **this**.responseText = **"You are not authorized"**;  }  }  };  }  });   *// Delete advert* $.mockjax(**function** (requestSettings) {  **if** (requestSettings.url.match(/https:\/\/mock\.api\.com\/appdata\/kid\_rk\/adverts\/(.+)/) &&  requestSettings.method === **"DELETE"**) {  **let** advertId = Number(requestSettings.url.match(/https:\/\/mock\.api\.com\/appdata\/kid\_rk\/adverts\/(.+)/)[1]);  **return** {  response: **function** (origSettings) {  **if** (requestSettings.headers[**"Authorization"**].includes(**"Kinvey mock\_token"**)) {  adverts = adverts.filter(a => a.\_id !== advertId);  **this**.responseText = adverts;  } **else** {  **this**.**status** = 403;  **this**.responseText = **"You are not authorized"**;  }  }  };  }  });    *// Edit advert* $.mockjax(**function** (requestSettings) {  **if** (requestSettings.url.match(/https:\/\/mock\.api\.com\/appdata\/kid\_rk\/adverts\/(.+)/) &&  requestSettings.method === **"PUT"**) {  **let** advertId = Number(requestSettings.url.match(/https:\/\/mock\.api\.com\/appdata\/kid\_rk\/adverts\/(.+)/)[1]);  **return** {  response: **function** (origSettings) {  **if** (requestSettings.headers[**"Authorization"**].includes(**"Kinvey mock\_token"**)) {  **let** advert = adverts.filter(a => a.\_id === advertId);  **let** data = requestSettings.data;  **if** (advert.**length** > 0) {  advert = advert[0];  advert.**title** = data.title;  advert.**publisher** = data.**publisher**;  advert.**datePublished** = data.**datePublished**;  advert.**price** = data.**price**;  **this**.responseText = advert;  }  **this**.responseText = {};  } **else** {  **this**.**status** = 403;  **this**.responseText = **"You are not authorized"**;  }  }  };  }  }); })(); |