

Worksheet: Event Based Programming

A logged in user would like to get the articles they authored using the `getArticles(userId)` request to fetch their articles. The request returns an object that has a list of articles that looks like the following:

```
{ articles: ["I like Rice", "What will be names of the 12th and 13th colleges"] }
```

Can you modify the code from slide 13 to safely get the user's articles? Why or why not?



Get Articles for logged in users with DDTs

```
var username = ...
var password = ...
...
var regUser = newDataDrivenFuture();
var logUser = newDataDrivenFuture();
var loggedIn = newDataDrivenFuture();
var logOut = newDataDrivenFuture();
...
async(() -> regUser.put(registerNewUser(username, password))); // { username: user, result: "success" or "failure"}
...
asyncAwait(regUser, () -> { if (regUser.safeGet().result.equals("success"))
    logUser.put(loginUser(username, password)); // {userId: id, result: "success" or "failure"}
    else
    logUser.put({result: "failure" }); });
...
asyncAwait(logUser, () -> { if (logUser.safeGet().result.equals("success"))
    loggedIn.put(isLoggedIn(logUser.safeGet().userId)); // {userId: id, result: "success" or "failure" }
    else
    loggedIn.put({result: "failure" }); });
...
asyncAwait(loggedIn, () -> { if (loggedIn.safeGet().result.equals("success"))
    logOut.put(logoutUser(loggedIn.safeGet().userId)); // { result: "success" or "failure" }
    else
    logOut.put({result: "failure" }); });
...
...

```



Get Articles for logged in users with DDTs

```
var username = ...
var password = ...
var regUser = newDataDrivenFuture();
var logUser = newDataDrivenFuture();
var loggedIn = newDataDrivenFuture();
var logOut = newDataDrivenFuture();
var articles = newDataDrivenFuture();
...
async(() -> regUser.put(registerNewUser(username, password))); // { username: user, result: "success" or "failure"}
...
asyncAwait(regUser, () -> { if (regUser.safeGet().result.equals("success"))
    logUser.put(loginUser(username, password)); // {userId: id, result: "success" or "failure"}
    else
    logUser.put({result: "failure" }); });
...
asyncAwait(logUser, () -> { if (logUser.safeGet().result.equals("success"))
    loggedIn.put(isLoggedIn(logUser.safeGet().userId)); // {userId: id, result: "success" or "failure" }
    else
    loggedIn.put({result: "failure" }); });
...
asyncAwait(loggedIn, () -> { if (loggedIn.safeGet().result.equals("success"))
    logOut.put(logoutUser(loggedIn.safeGet().userId)); // { result: "success" or "failure" }
    else
    logOut.put({result: "failure" }); });
...
asyncAwait(loggedIn, () -> { if (loggedIn.safeGet().result.equals("success"))
    articles.put(getArticles(loggedIn.safeGet().userId)); // { articles: ["a1", "a2", "a3"] }
    else
    articles.put({articles: [] }); });
...
...

```



Get Articles for logged in users with DDTs

```
var username = ...
var password = ...
var regUser = newDataDrivenFuture();
var logUser = newDataDrivenFuture();
var loggedIn = newDataDrivenFuture();
var logOut = newDataDrivenFuture();
var articles = newDataDrivenFuture();
...
async(() -> regUser.put(registerNewUser(username, password))); // { username: user, result: "success" or "failure"}
...
asyncAwait(regUser, () -> { if (regUser.safeGet().result.equals("success"))
    logUser.put(loginUser(username, password)); // {userId: id, result: "success" or "failure"}
    else
    logUser.put({result: "failure" }); });
...
asyncAwait(logUser, () -> { if (logUser.safeGet().result.equals("success"))
    loggedIn.put(isLoggedIn(logUser.safeGet().userId)); // {userId: id, result: "success" or "failure" }
    else
    loggedIn.put({result: "failure" }); });
...
asyncAwait(loggedIn, () -> { if (loggedIn.safeGet().result.equals("success"))
    logOut.put(logoutUser(loggedIn.safeGet().userId)); // { result: "success" or "failure" }
    else
    logOut.put({result: "failure" }); });
...
asyncAwait(loggedIn, () -> { if (loggedIn.safeGet().result.equals("success"))
    articles.put(getArticles(loggedIn.safeGet().userId)); // { articles: ["a1", "a2", "a3"] }
    else
    articles.put({articles: [] }); });
...
...

```

Not safe. What happens if the logout task completes before the get article task?



Get Articles for logged in users with DDTs

```
var username = ...
var password = ...
var session = -1;
var regUser = newDataDrivenFuture();
var logUser = newDataDrivenFuture();
var loggedIn = newDataDrivenFuture();
var logOut = newDataDrivenFuture();
var articles = newDataDrivenFuture();
...
async(() -> regUser.put(registerNewUser(username, password))); // { username: user, result: "success" or "failure"}
...
asyncAwait(regUser, () -> { if (regUser.safeGet().result.equals("success"))
    logUser.put(loginUser(username, password)); // {userId: id, result: "success" or "failure"}
    else
    logUser.put({result: "failure" }); });
...
asyncAwait(logUser, () -> { if (logUser.safeGet().result.equals("success"))
    session = logUser.safeGet().userId; loggedIn.put(isLoggedIn(session)); // {userId: id, result: "success" or "failure" }
    else
    loggedIn.put({result: "failure" }); });
...
asyncAwait(loggedIn, () -> { if (loggedIn.safeGet().result.equals("success"))
    logOut.put(logoutUser(session)); session = -1; // { result: "success" or "failure" }
    else
    logOut.put({result: "failure" }); });
...
asyncAwait(loggedIn, () -> { if (loggedIn.safeGet().result.equals("success") && session != -1)
    articles.put(getArticles(session)); // { articles: ["a1", "a2", "a3"]}
    else
    articles.put({articles: [] }); });
...

```

Still not entirely safe. What happens if logoutUser() is called right before getArticles()? We should move isLoggedIn() to the server and have logoutUser() and getArticles() internally call isLoggedIn().

