

App Studio 4.0 Deployment Guide

2019-10-18

Table of Contents

- Deployment Overview 1
 - Servlet container or Fusion Web Apps service 1
 - Standalone deployments 1
- Fusion Web App Deployment 2
 - UI deployment overview 3
 - Publishing a search interface to Fusion Server 4
 - Migrating a search interface to other Fusion hosts 6
 - Download or build a WAR file 6
 - Downloading a WAR file 6
 - Building a WAR file 8
 - Upload the WAR file 9
 - Uploading with the Fusion UI 9
 - Uploading with the REST API 11
 - Running a dedicated node for a search interface 12
- Standalone Mode 13
 - Standalone mode installation 14
 - Standalone mode startup 15
 - Enabling production mode 17
- SSL Configuration 18
 - SSL keystore 19
 - SSL parameters 20
- Enabling Social Features 21
 - Configure Fusion 22
 - Configure App Studio 24
 - Upgrade to the latest version of Appkit 24
 - Enable social features 24

Deployment Overview

You can choose how to deploy your search application:

- **Servlet container or Fusion Web Apps service** – Deploy your search application in a servlet container or in the Fusion Web Apps service.

Tip	This is the recommended deployment method for production.
-----	---

- Deploy your search application as a standalone application.

Servlet container or Fusion Web Apps service

Deploy your search application in a servlet container:

- Java WAR file for deployment in a servlet container

Package your search app in a Java WAR file, which can be deployed in a servlet container such as Tomcat or Jetty. This is the most common way to deploy a search app.

- Fusion Web app deployment gives you an embedded instance of App Studio in each of your Fusion instances.

Standalone deployments

Two types of standalone deployments are supported:

- Java JAR file with embedded Tomcat web server

Package your search app in a Java JAR file that also contains an embedded Tomcat web server. The app is ready to deploy in a production environment. It isn't necessary to build a WAR file and deploy that in a servlet container or application server.

- Executable Java JAR file for execution in a Java runtime environment

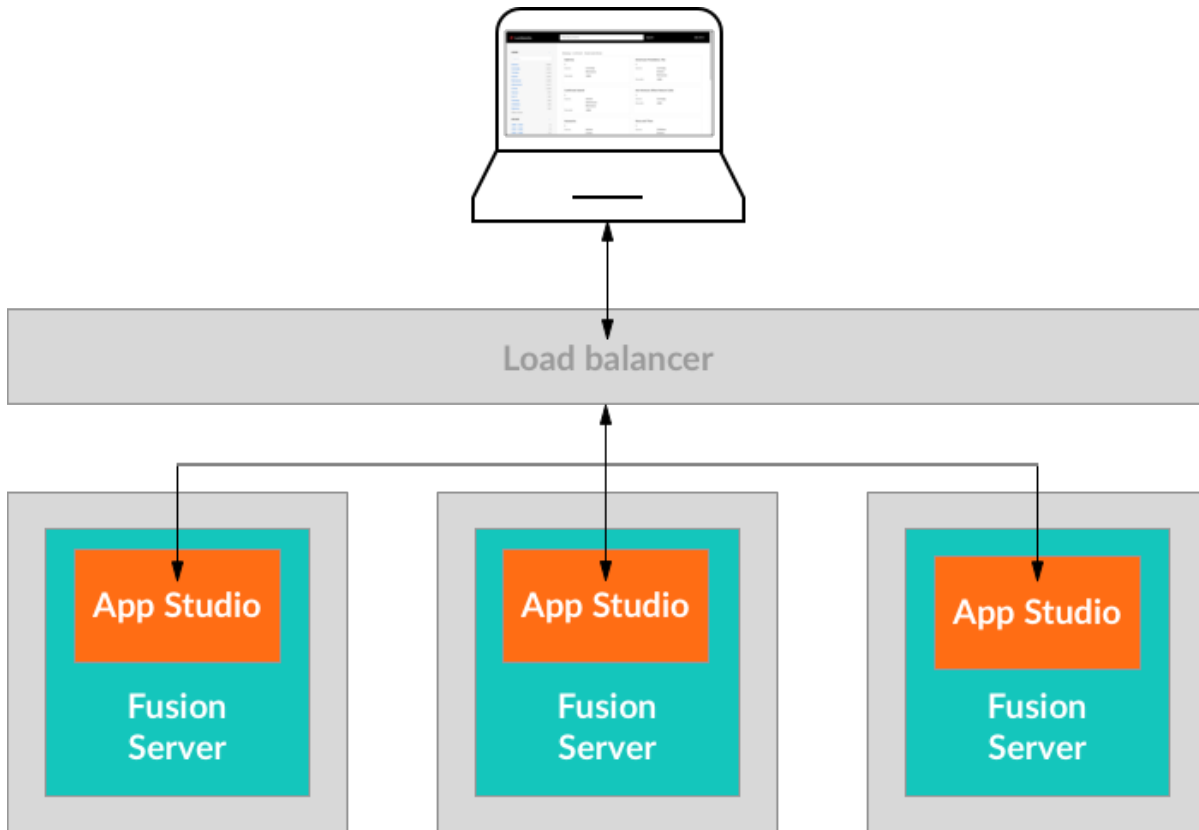
Package your search app in a Java JAR file. You can run the app with the Java launcher.

Fusion Web App Deployment

You can use App Studio to deploy a search app in Fusion Server. You develop your search interface, and then publish the search app to Fusion's Web Apps service.

Note	This deployment type leverages your existing production environment, requiring no dedicated nodes for App Studio. However, you might need to deploy additional Fusion nodes if you anticipate significant traffic to your search interface.
------	---

UI deployment overview



Here's an overview of the steps:

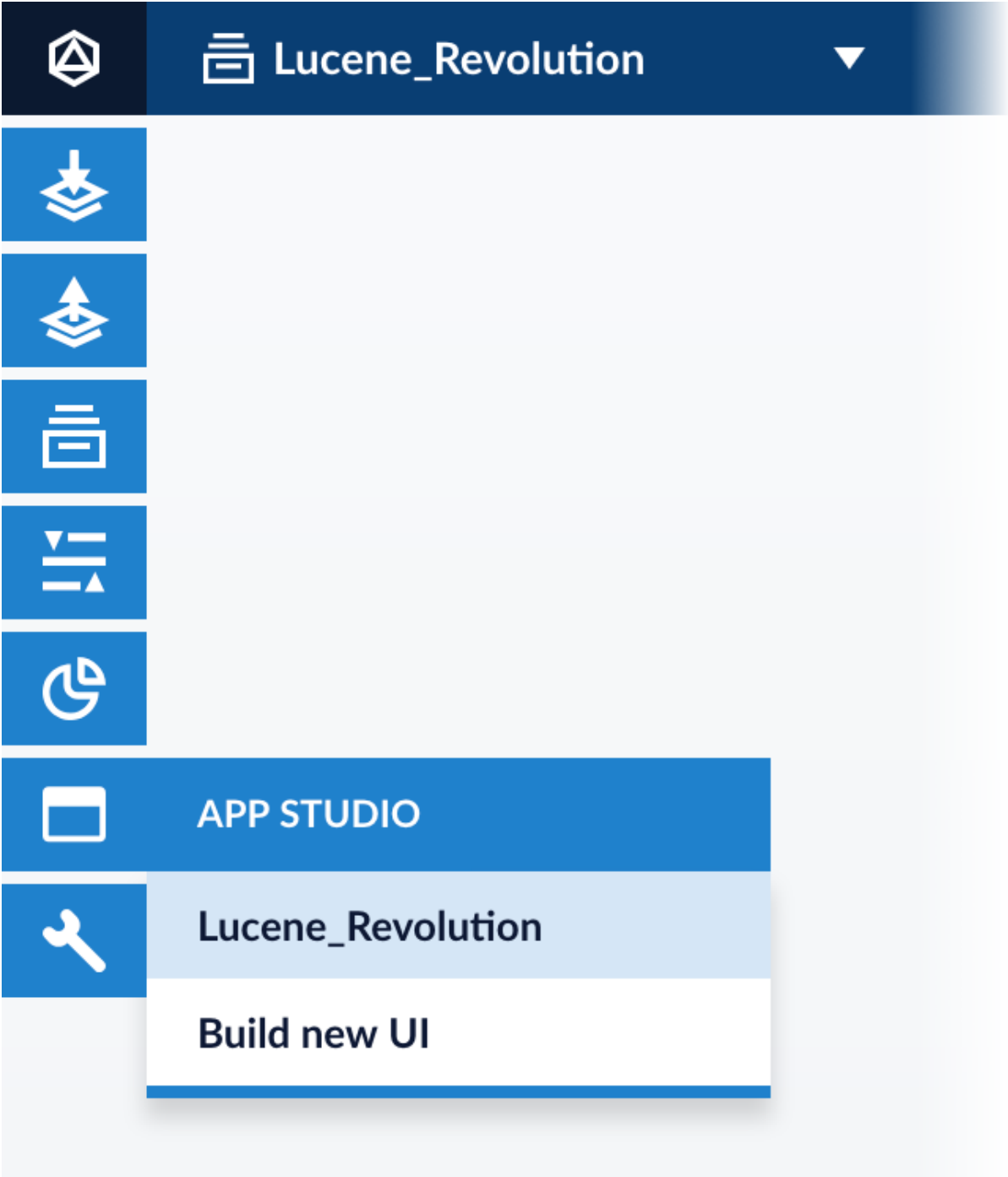
1. Develop your search interface.
2. Publish your UI to Fusion Server using the instructions below.

Your web app then becomes accessible at `<a href="http://localhost:8764/webapps/<project-name>" class="bare">http://localhost:8764/webapps/<project-name>`.

Publishing a search interface to Fusion Server

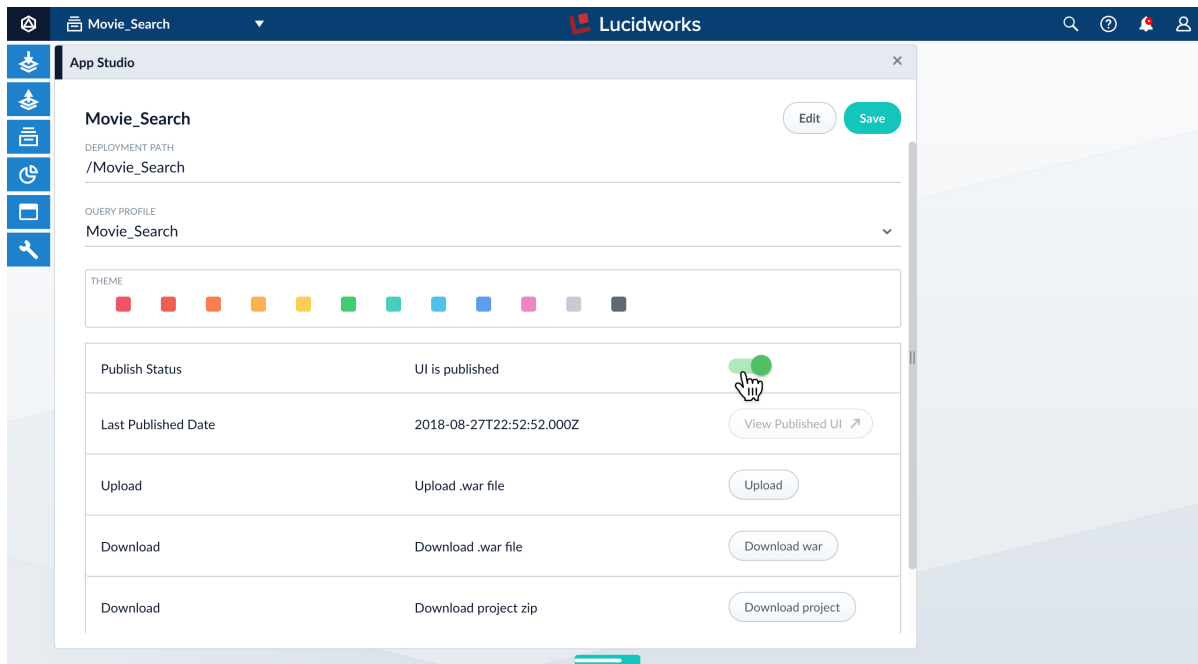
These steps publish your search interface to a Fusion instance (and its cluster) where the App Studio project already exists.

- 1. In the Fusion workspace, navigate to **App Studio** and select your search interface.



The configuration panel appears.

- 2. Next to **Publish Status**, click the toggle:



3. Click **Save**.

The interface is only published after you click **Save**, which also enables the **View Published UI** button.

4. Click **View Published UI** to launch a new window where you can view the interface as an end user.

Your published interface is available at <http://localhost:8764/webapps/<project-name>#/search>.

Migrating a search interface to other Fusion hosts

If you've published your search interface on one Fusion Host in a cluster, then it is published throughout the cluster.

There are two ways to migrate a search interface to a different Fusion cluster:

Export and import the Fusion app	Download and upload the WAR (.war) file
<p>This migrates all objects in the Fusion app, such as datasources, query profiles, schedules, and so on, in addition to the search interface.</p> <p>The search interface will be editable on the target hosts.</p> <p>For instructions, see Working With Apps in Fusion Server's Getting Started guide.</p>	<p>This migrates only the search interface. The search interface will not be editable on the target hosts.</p> <p>If you are uploading the file into a Fusion app other than the one in which it was created, the new Fusion app must include a query profile and data fields whose names are identical to the ones already configured in the search interface.</p> <p>For instructions on downloading or building a WAR file and uploading it to Fusion, see below.</p>

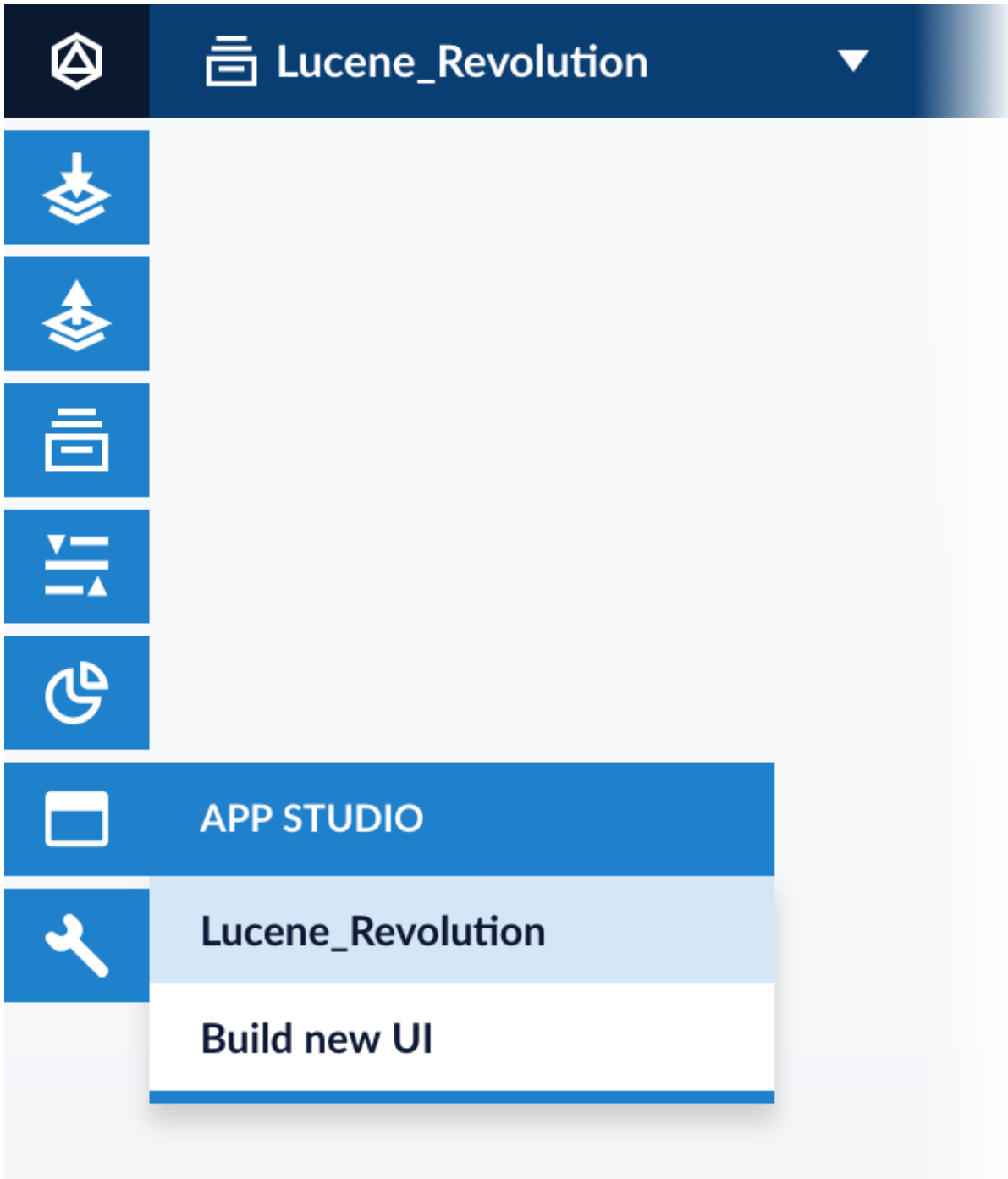
Download or build a WAR file

You can distribute a search interface as a WAR (.war) file.

You can download a WAR file from the Fusion UI or create one from a downloaded project.

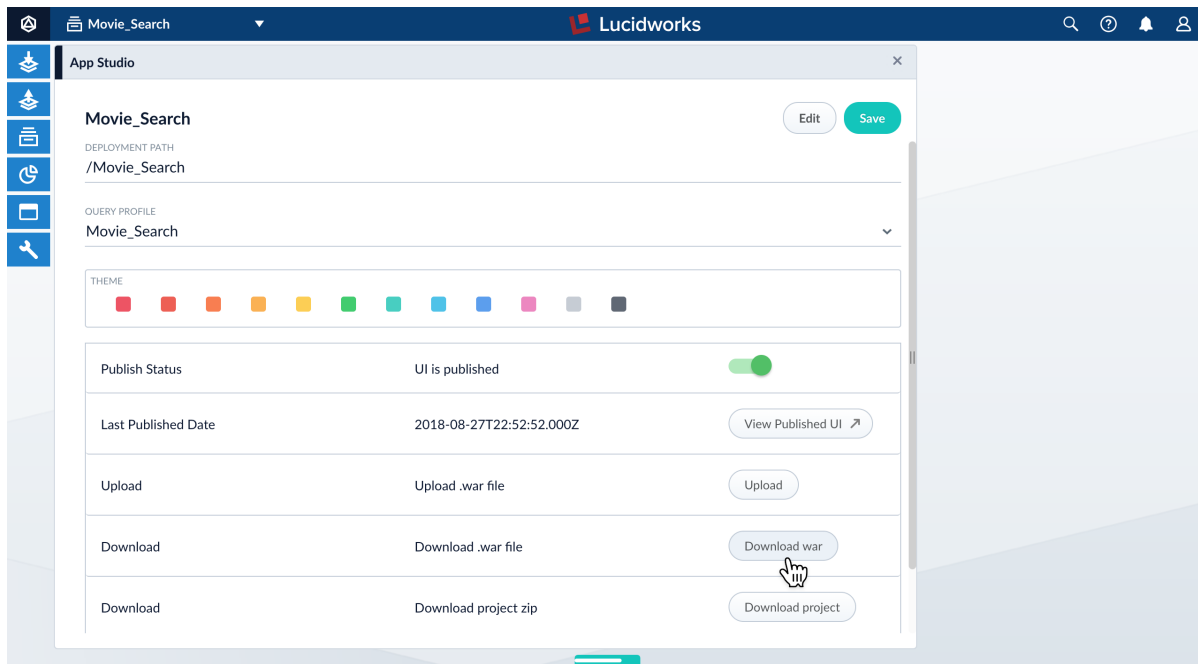
Downloading a WAR file

1. In the Fusion workspace, navigate to **App Studio** and select your search interface.



The configuration panel appears.

2. Click **Download war**:



Building a WAR file

A downloaded project comes with the scripts `app-studio` (for Unix) and `app-studio.bat` (for Windows) that create WAR files.

How to build a WAR file

Unix and MacOS:

1. In a shell window, switch to your project directory:

```
cd /path/to/project-directory
```

2. Create application files:

```
./app-studio package
```

Windows:

1. In a terminal window, switch to your project directory:

```
cd \path\to\project-directory
```

2. Create application files:

```
app-studio.bat package
```

Application files created:

The script creates the following files in the `dist` directory of your project:

- `search-app-4.2.x.jar`

This is a executable Java JAR file file.

- `search-app-project.zip`

Use this file to share your project with other search interface developers.

- `search-app.war`

You can upload this file to Fusion to deploy it. See the Deployment Guide.

Upload the WAR file

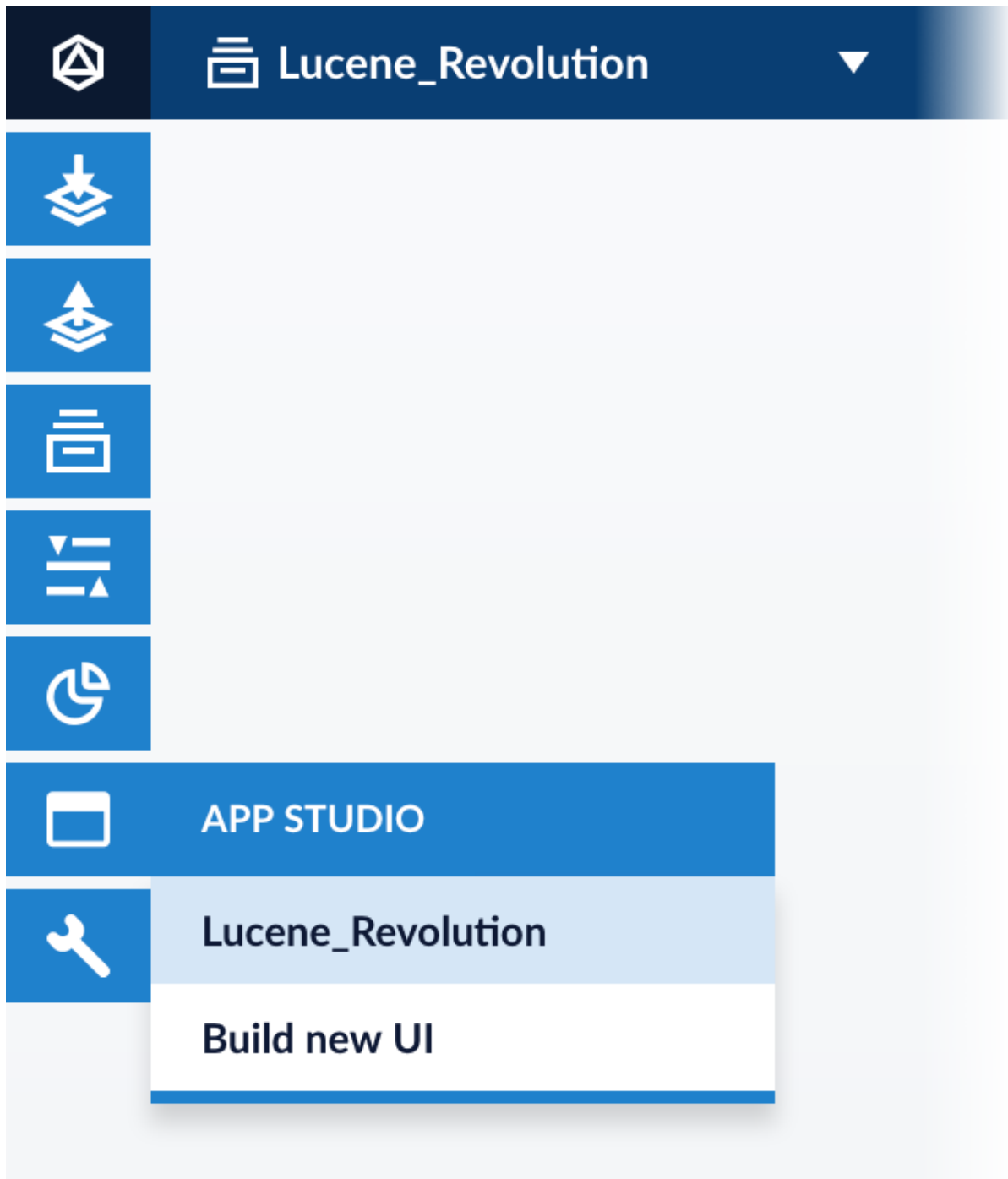
You can upload the WAR file through the Fusion UI or the REST API.

Uploading with the Fusion UI

You can upload your project (as a WAR file) in the App Studio configuration panel.

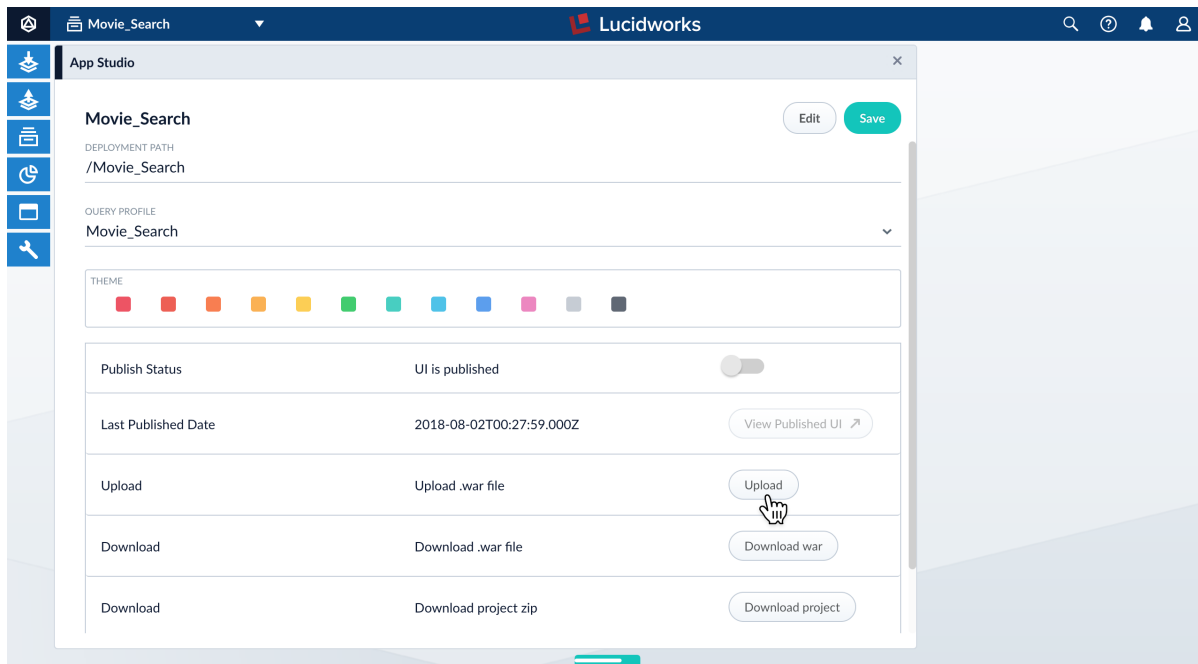
Note	If you upload the file into a Fusion app other than the one in which it was created, the new Fusion app must include a query profile and data fields whose names are identical to the ones already configured in the search interface.
------	--

1. From the App Studio menu, select your project:



The project configuration panel appears.

2. Click **Upload**.



Fusion prompts you to confirm that you want to upload a project that will delete the existing one.

3. Click **Yes, continue**.
4. Navigate to your WAR file and select it.

After the upload is complete, you can edit or publish the uploaded project.

Uploading with the REST API

Use the `webapps/{id}/war` endpoint of the Webapps API to upload a WAR file, as in this example:

```
curl -u admin:password123 -X PUT http://localhost:8764/api/webapps/Movie_Search/war -F 'file=@movies.war'
```

Running a dedicated node for a search interface

When deploying Fusion nodes dedicated to serving one or more search interfaces, only the following Fusion services need to run:

- `agent`
- `api`
- `log-shipper`
- `webapps`

When starting a dedicated node, you can minimize the number of running services by starting it like this.

Unix and MacOS:

```
cd fusion/4.2.x/bin
./agent start
./api start
./log-shipper start
./webapps start
```

Windows:

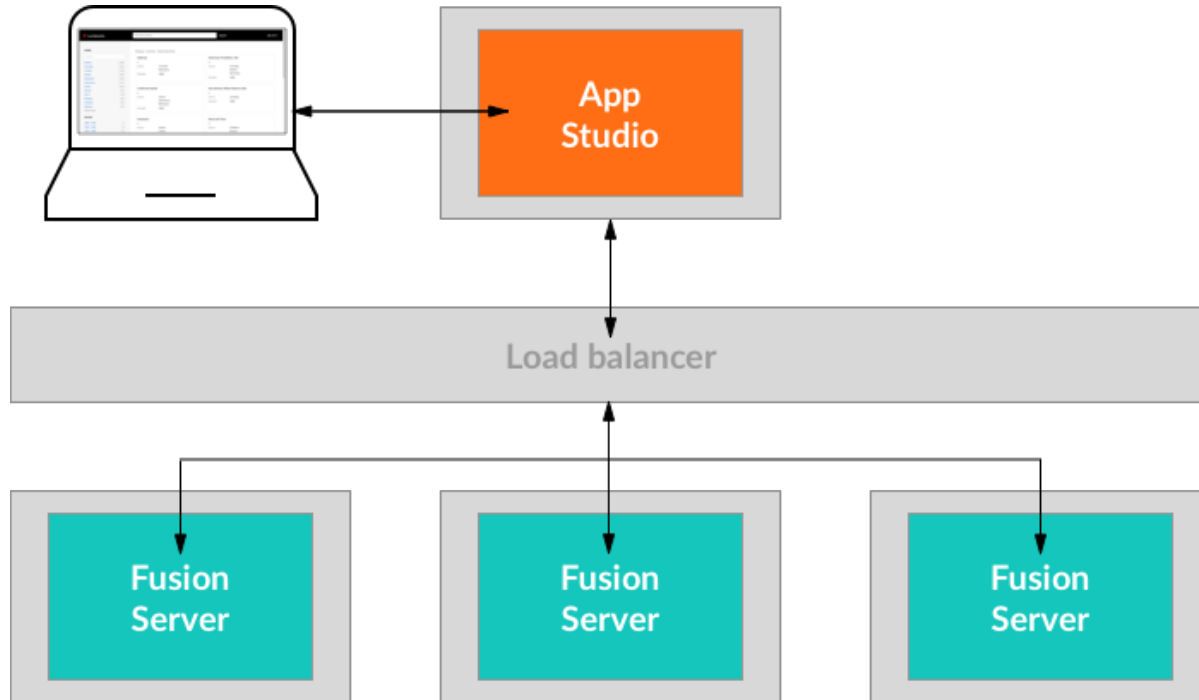
```
cd fusion\4.2.x\bin
agent.cmd start
api.cmd start
log-shipper.cmd start
webapps.cmd start
```

Standalone Mode

Standalone mode is what you get when you run the `app-studio` or `app-studio.bat` script. This deployment type is useful for:

- Evaluating App Studio
- Developing and demonstrating your Web apps
- Small production environments (see Enabling production mode below)

Standalone deployment



Standalone mode installation

1. [Download App Studio](#).
2. Copy the App Studio zip file to any convenient path.
3. Expand the archive: `unzip app-studio-4.2.x-zip-package.zip`

This creates an `app-studio` directory.

Standalone mode startup

These commands start App Studio in its default mode, for development and demos. To start App Studio in production mode, see Enabling production mode below.

Run App Studio at the command line from the `app-studio` directory, like this:

	Unix	Windows
Start the server	<pre>./app-studio.sh start</pre> <p>Options:</p> <ul style="list-style-type: none">• <code>--production</code> Run the server in production mode.• <code>--extraJvmArguments</code> Space character-delimited additional JVM arguments; multiple arguments must be quoted. Example: <code>"-Xms2G -Xmx5G"</code>• <code>-p</code> or <code>--port</code> The port on which to run the server.• <code>-s</code> or <code>--stop-port</code> The port to use when issuing the <code>stop</code> command; it must be re-specified when running <code>stop</code>.	<pre>app-studio-start.bat</pre>
Stop the server	<pre>./app-studio.sh stop</pre> <p>Optionally:</p> <ul style="list-style-type: none">• <code>-s</code> or <code>--stop-port</code> The port to use when issuing the <code>stop</code> command.	<pre>app-studio-standalone-stop.bat</pre>
Restart the server	<pre>./app-studio.sh restart</pre>	<pre>app-studio-stop.bat app-studio-start.bat</pre>
Get the server status	<pre>./app-studio.sh status</pre>	

You can run multiple instances of App Studio on a single host by specifying different ports for all instances. Do this on

startup, at the command line, using the `-p` parameter, for example:

```
./app-studio.sh start -p 9000
```

When running multiple instances, also use the `-p` flag to indicate which instance of App Studio to shut down, for example:

```
./app-studio.sh stop -p 9000
```

Enabling production mode

Production mode disables the configuration wizard and the code editor.

To use standalone mode in a production environment, start App Studio with the `--production` flag:

```
./app-studio-standalone.sh start --production
```

SSL Configuration

App Studio can be served over HTTPS using SSL encryption. You can use the default keystore for development and testing, or use your own keystore for production. Then, invoke the App Studio startup script using the SSL parameters.

SSL keystore

We include a keystore file with a default self-signed key for development and testing.

For proper security in a production environment, import your own keystore into the `keystore.jks` file, or copy it to a new file. If you copy it to a new file, use the `-Dtwikit.keystore.file` (described below) to specify its location.

SSL parameters

To enable SSL, you specify the following parameters on the command line when invoking the startup script:

Parameter	Description	Default
<code>-Dtwigkit.ssl=true</code>	Enable SSL.	<code>false</code>
<code>-Dtwigkit.https.port</code>	Set the port.	<code>8765</code>
<code>-Dtwigkit.keystore.file</code>	The keystore path/filename, relative to the <code>app-studio</code> directory.	<code>keystore.jks</code>
<code>-Dtwigkit.keystore.password</code>	The keystore password.	<code>p4ssw0rd</code>
<code>-Dtwigkit.keystore.alias</code>	The name of the key in the keystore to be used.	<code>default-key</code>

Enabling Social Features

Prior to using social and collaboration tools, you must enable them in Fusion and in App Studio.

Configure Fusion

How to configure Fusion

1. Update the `managed-schema` to include the new social fields:

```
<field indexed="true" multiValued="false" name="type" required="false" stored="true" type="string"/>
  <field indexed="true" multiValued="false" name="user_id" required="false" stored="true" type="string"/>
  <field indexed="true" multiValued="false" name="full_name" required="false" stored="true"
type="string"/>
  <field indexed="true" multiValued="false" name="anonymous" required="false" stored="true"
type="boolean"/>
  <field indexed="true" multiValued="false" name="created" required="false" stored="true" type="pdate"/>
  <field indexed="true" multiValued="false" name="created_epoch" required="false" stored="true"
type="plong"/>
  <field indexed="true" multiValued="false" name="name" required="false" stored="true" type="string"/>
  <field indexed="true" multiValued="false" name="path" required="false" stored="true" type="string"/>
  <field indexed="true" multiValued="false" name="private" required="false" stored="true" type="boolean"/>
  <field indexed="true" multiValued="false" name="query_url" required="false" stored="true"
type="string"/>
  <field indexed="true" multiValued="false" name="collection" required="false" stored="true"
type="string"/>
  <field indexed="true" multiValued="false" name="target" required="false" stored="true" type="string"/>
  <field indexed="true" multiValued="false" name="text" required="false" stored="true" type="string"/>
  <field indexed="true" multiValued="false" name="title" required="false" stored="true" type="string"/>
  <field indexed="true" multiValued="false" name="description" required="false" stored="true"
type="string"/>
  <field indexed="true" multiValued="false" name="url" required="false" stored="true" type="string"/>
  <field indexed="true" multiValued="false" name="accessibility" required="false" stored="true"
type="string"/>
  <field indexed="true" multiValued="false" name="namespace" required="false" stored="true"
type="string"/>
  <field indexed="true" multiValued="false" name="creator__anonymous" required="false" stored="true"
type="boolean"/>
  <field indexed="true" multiValued="false" name="creator__full_name" required="false" stored="true"
type="string"/>
  <field indexed="true" multiValued="false" name="creator__id" required="false" stored="true"
type="plong"/>
  <field indexed="true" multiValued="false" name="creator__user_id" required="false" stored="true"
type="string"/>
  <field indexed="true" multiValued="false" name="topic__id" required="false" stored="true" type="plong"/>
  <field indexed="true" multiValued="false" name="topic__title" required="false" stored="true"
type="string"/>
  <field indexed="true" multiValued="false" name="topic__private" required="false" stored="true"
type="boolean"/>
  <field indexed="true" multiValued="false" name="topic__path" required="false" stored="true"
type="string"/>
  <field indexed="true" multiValued="false" name="topic__namespace" required="false" stored="true"
type="string"/>
  <field indexed="true" multiValued="false" name="topic__description" required="false" stored="true"
type="string"/>
  <field indexed="true" multiValued="false" name="topic__accessibility" required="false" stored="true"
type="string"/>
  <field indexed="true" multiValued="false" name="topic__collection" required="false" stored="true"
type="string"/>
  <field indexed="true" multiValued="false" name="topic__created" required="false" stored="true"
```



```
type="pdate"/>
  <field indexed="true" multiValued="false" name="topic__created_epoch" required="false" stored="true"
type="plong"/>
  <field indexed="true" multiValued="false" name="topic__creator__anonymous" required="false"
stored="true" type="boolean"/>
  <field indexed="true" multiValued="false" name="topic__creator__full_name" required="false"
stored="true" type="string"/>
  <field indexed="true" multiValued="false" name="topic__creator__id" required="false" stored="true"
type="plong"/>
  <field indexed="true" multiValued="false" name="topic__creator__user_id" required="false" stored="true"
type="string"/>
```

Configure App Studio

Upgrade to the latest version of Appkit

Upgrade to the latest version of Appkit.

Enable social features

1. Create the file `social.conf` in the `conf/social` directory and add the following parameter to it:

```
platform: platforms.fusion.social
```

2. Create the file `social.conf` in the `conf/platforms/Fusion` directory and add the following parameters to it:

```
# Required for query/index pipeline
collection: fusion-app-name_user_data

# Required for query/index profiles
index-profile: fusion-app-name_user_data
query-profile: fusion-app-name_user_data

# Allow Fusion platform to store Social entities
readOnly: false
webservice-enabled: false
```