

IBM Cloud App ID (доработанный шаблон) для аутентификации чатботом.

Назначение приложения

Обработка аутентификации

`app.js` Uses Express - роутеры .

`public/index.html` Стартовая страница прилоения.

`protected/protected.html` - Защищенные ресурсы.

Настройки

Настройки описаны в файле `./mappings.json`

Запуск локально

В корневом каталоге положить файл с настройками: `localdev-config.json`

```
{
  "clientId": "<clientId>",
  "tenantId": "tenantId",
  "secret": "<secret>",
  "name": "node-app",
  "oAuthServerUrl": "https://",
  "profilesUrl": "https://",
  "discoveryEndpoint": "https://",
  "type": "regularwebapp",
  "scopes": [],
  "redirectUri": "http://"
}
```

Запустить команду:

```
npm install
npm start
```

Вход в приложение через browser `http://localhost:8080` .

Запуск в Cloud Foundry

Заполнить env переменные и парметры application в `manifest.yml`

Deployment

Важно: Для запуска удалить `http://localhost:8080/*` со списка web redirect URLs, что "Manage Authentication" -> "Authentication Settings" вкладке AppID dashboard.

1. Login to IBM Cloud.

```
ibmcloud login -a https://api.{{domain}}
```

2. Target a Cloud Foundry organization and space in which you have at least Developer role access:

Use `ibmcloud target --cf` to target Cloud Foundry org/space interactively.

3. Bind the sample app to the instance of App ID:

```
ibmcloud resource service-alias-create "appIDInstanceName-alias" --instance-name  
"appIDInstanceName" -s {{space}}
```

4. Add the alias to the manifest.yml file in the sample app.

5. Deploy the sample application to IBM Cloud. From the app's folder do:

```
ibmcloud app push
```

6. Добавить новый redirect URL в App ID dashboard. Для этого, в вашем App ID instance зайти в Manage Authentication->Authentication Settings->Add web redirect URLs и добавить новый URL:

```
https://{App Domain}/ibm/cloud/appid/callback
```

You find your app's domain by visiting Cloud Foundry Apps at the IBM Cloud dashboard:
`https://cloud.ibm.com/resources`.

7. Open your IBM Cloud app route in the browser.

Running in Kubernetes

Prerequisites

Before you begin make sure that IBM Cloud CLI, docker and kubectl installed and that you have a running kubernetes cluster. You also need an IBM Cloud container registry namespace (see <https://cloud.ibm.com/kubernetes/registry/main/start>). You can find your registry domain and repository namespace using `ibmcloud cr namespaces`.

Deployment

Important: Before going live, remove `http://localhost:3000/*` from the list of web redirect URLs located in "Manage Authentication" -> "Authentication Settings" page in the AppID dashboard.

Note: Your App ID instance name must consist of lower case alphanumeric characters, '-' or '.', and must start and end with an alphanumeric character. You can visit the App ID dashboard to change your instance name.

1. Login to IBM Cloud.

```
ibmcloud login -a https://api.{{domain}}
```

2. Run the following command, it will output an export command.

```
ibmcloud cs cluster-config {CLUSTER_NAME}
```

3. Set the KUBECONFIG environment variable. Copy the output from the previous command and paste it in your terminal. The command output looks similar to the following example:

```
export KUBECONFIG=/Users/$USER/.bluemix/plugins/container-  
service/clusters/mycluster/kube-config-hou02-mycluster.yml
```

4. Bind the instance of App ID to your cluster.

```
ibmcloud cs cluster-service-bind {CLUSTER_NAME} default {APP_ID_INSTANCE_NAME}
```

5. Find your cluster's public endpoint {CLUSTER_ENDPOINT}.

Note: If you are using the free version of kubernetes (with only 1 worker node) you can use your node's public IP instead, which you can find using:

```
ibmcloud cs workers {CLUSTER_NAME}
```

6. Edit the kube_deployment.yml file.

1. Edit the image field of the deployment section to match your image name. The name of your image should be {REGISTRY_DOMAIN}/{REPOSITORY_NAMESPACE}/appid-node-sample:{APP_VERSION}).
2. Edit the Binding name field to match yours. It should be binding-{APP_ID_INSTANCE_NAME}.
3. Edit redirectUri's value to include your cluster's IP. The value should be `http://{CLUSTER_ENDPOINT}/ibm/cloud/appid/callback`
4. Optional: Change the value of metadata.namespace from default to your cluster namespace if you're using a different namespace.

7. Build your Docker image.

```
docker build -t {REGISTRY_DOMAIN}/{REPOSITORY_NAMESPACE}/appid-node-sample:  
{APP_VERSION} .
```

8. Push the image.

```
docker push {REGISTRY_DOMAIN}/{REPOSITORY_NAMESPACE}/appid-node-sample:  
{APP_VERSION}
```

```
kubectl apply -f kube_deployment.yml
```

9. Now configure the OAuth redirect URL at the App ID dashboard so it will approve redirecting to your cluster. Go to your App ID instance at [IBM Cloud console](#) and under Manage Authentication->Authentication Settings->Add web redirect URLs add the following URL:

```
https://{CLUSTER_ENDPOINT}:30000/ibm/cloud/appid/callback
```

10. You can see your sample running on Kubernetes in IBM Cloud.

```
open http://{CLUSTER_ENDPOINT}:30000
```

Clarification

This sample runs on one instance and uses the session to store the authorization data. In order to run it in production mode, use services such as Redis to store the relevant data.

Полезные ресурсы

Protecting Node.js Web Applications with IBM Cloud App ID

<https://www.youtube.com/watch?v=6roa1ZOvwtw>

https://eu-gb.appid.cloud.ibm.com/oauth/v4/c321f792-f62c-42d5-8537-d311f3f69db4/authorization?client_id=ca1b7c1b-4f16-4927-9723-3227a7211f27&response_type=code&redirect_uri=http://localhost:8080/ibm/cloud/appid/callback&scope=appid_default&language=en-US&state=dsy7Vvm73gdwHjHiUoFfSYCjk3c=&language=en-US