

Configurations Onlyoffice

Voici la liste des fichiers de configuration des plateformes onlyoffice dans les établissements, vous pouvez ajouter le votre à la suite.

Université de Lorraine



Depuis la version 7.2, Onlyoffice refuse de prendre en compte le fichier mappé local.json, puisque c'est un fichier qui est généré au démarrage du conteneur.

Néanmoins, il est possible de définir les options que l'on souhaite, en les surchargeant dans un nouveau fichier **local-production-linux.json**, qu'OnlyOffice prendra en compte lors de la génération du fichier local.json

local-production-linux.json

```
{
  "services": {
    "CoAuthoring": {
      "secret": {
        "inbox": {
          "string": "password"
        },
        "outbox": {
          "string": "password"
        },
        "session": {
          "string": "password"
        }
      },
      "autoAssembly": {
        "enable": true,
        "interval": "180m",
        "step": "60m"
      },
      "expire": {
        "sessionidle": "120m",
      }
    }
  },
  "license": {
    "license_file": "/var/www/onlyoffice/Data/license.lic"
  },
  "FileConverter": {
    "converter": {
      "downloadTimeout": 240
    }
  }
}
```

docker-compose.yml

```
version: '2'
#Définition des services
services:
  onlyoffice-document-server:
    container_name: onlyoffice-ds
    image: onlyoffice/documentserver-ee:6.2
    restart: always
    volumes:
      - /applications/onlyoffice/data/logs:/var/log/onlyoffice/documentserver
      - /applications/onlyoffice/data/lib:/var/lib/onlyoffice
      - /applications/onlyoffice/data/data:/var/www/onlyoffice/Data
      - /applications/onlyoffice/data/db:/var/lib/postgresql
      - /applications/onlyoffice/data/local-production-linux.json:/etc/onlyoffice/documentserver/local-
production-linux.json
      - /applications/onlyoffice/data/themes:/var/www/onlyoffice/documentserver/sdkjs/slide/themes/src
```

Université de Grenoble

local.json

```
{
  "services": {
    "CoAuthoring": {
      "expire": {
        "sessionidle": "10m",
        "files": 900
      },
      "sql": {
        "dbHost": "BDD FQDN",
        "dbName": "DBNAME",
        "dbUser": "*****",
        "dbPass": "*****",
        "type": "postgres",
        "dbPort": "*****"
      },
      "redis": {
        "host": "REDIS FQDN"
      },
      "token": {
        "enable": {
          "request": {
            "inbox": false,
            "outbox": false
          },
          "browser": false
        },
        "inbox": {
          "header": "Authorization"
        },
        "outbox": {
          "header": "Authorization"
        }
      },
      "secret": {
        "inbox": {
          "string": "*****"
        },
        "outbox": {
          "string": "*****"
        },
        "session": {
          "string": "*****"
        }
      }
    },
    "rabbitmq": {
      "url": "amqp://USER:PASS@REDIS FQDN"
    }
  }
}
```

local.json

```
{
  "services": {
    "CoAuthoring": {
      "sql": {
        "type": "postgres",
        "dbHost": "[HOST]",
        "dbPort": "[PORT]",
        "dbName": "[NAME]",
        "dbUser": "[USER]",
        "dbPass": "[PASS]"
      },
      "redis": {
        "host": "[HOST]"
      },
      "autoAssembly": {
        "enable": true,
        "interval": "5m"
      },
      "expire": {
        "sessionidle": "30m"
      },
      "token": {
        "enable": {
          "request": {
            "inbox": false,
            "outbox": false
          },
          "browser": false
        },
        "inbox": {
          "header": "Authorization"
        },
        "outbox": {
          "header": "Authorization"
        }
      },
      "secret": {
        "inbox": {
          "string": "[SECRET]"
        },
        "outbox": {
          "string": "[SECRET]"
        },
        "session": {
          "string": "[SECRET]"
        }
      }
    }
  },
  "rabbitmq": {
    "url": "amqp://[USER]:[PASS]@[HOST]"
  }
}
```