

















Guida Test Smells Game

Installazione Server

È possibile installare l'infrastruttura server in locale scaricando il repository e navigando nella directory "Test-Smells-Game/Projects/Server-Setup". In questa cartella bisogna aprire un terminale e digitare il seguente comando:

docker-compose up -d

In questo modo verranno creati tutti i container necessari:

<input type="checkbox"/>	▼		server-setup		Running (5/5)
<input type="checkbox"/>			exercise-service cd8a287d23c7 	loriszn/server-setup:exercis	Running 9090:9090 
<input type="checkbox"/>			leaderboard-servi ca6f8ace5601 	loriszn/server-setup:leaderb	Running 8087:8087 
<input type="checkbox"/>			user-service b95f5e1d3dbd 	loriszn/server-setup:user_s	Running 8081:8081 
<input type="checkbox"/>			frontend 8e496dd41b05 	loriszn/server-setup:fronter	Running 4200:80 
<input type="checkbox"/>			compiler-service 5715b27939b9 	loriszn/server-setup:compil	Running 8083:8083 

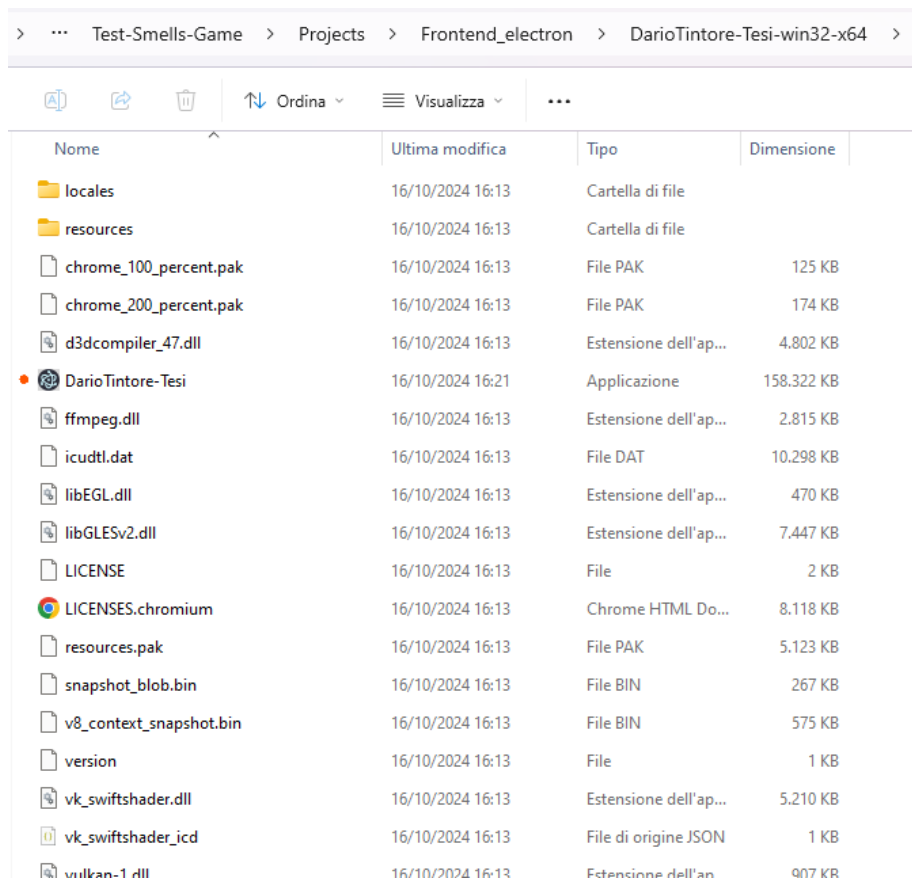
Dopo l'avvio dei container, sarà possibile accedere al sistema tramite il frontend al seguente indirizzo: **http://localhost:4200/**.

Installazione Client Electron

Per installare il client Electron bisogna recarsi nella directory "Test-Smells-Game/Projects/Frontend_electron" e aprire un terminale, digitando il comando

npm install

Al termine dell'installazione verrà aggiunta una nuova directory all'interno del progetto in cui sarà presente un eseguibile che rappresenta il Client Electron.



Nome	Ultima modifica	Tipo	Dimensione
locales	16/10/2024 16:13	Cartella di file	
resources	16/10/2024 16:13	Cartella di file	
chrome_100_percent.pak	16/10/2024 16:13	File PAK	125 KB
chrome_200_percent.pak	16/10/2024 16:13	File PAK	174 KB
d3dcompiler_47.dll	16/10/2024 16:13	Estensione dell'ap...	4.802 KB
DarioTintore-Tesi	16/10/2024 16:21	Applicazione	158.322 KB
ffmpeg.dll	16/10/2024 16:13	Estensione dell'ap...	2.815 KB
icudtl.dat	16/10/2024 16:13	File DAT	10.298 KB
libEGL.dll	16/10/2024 16:13	Estensione dell'ap...	470 KB
libGLESv2.dll	16/10/2024 16:13	Estensione dell'ap...	7.447 KB
LICENSE	16/10/2024 16:13	File	2 KB
LICENSES.chromium	16/10/2024 16:13	Chrome HTML Do...	8.118 KB
resources.pak	16/10/2024 16:13	File PAK	5.123 KB
snapshot_blob.bin	16/10/2024 16:13	File BIN	267 KB
v8_context_snapshot.bin	16/10/2024 16:13	File BIN	575 KB
version	16/10/2024 16:13	File	1 KB
vk_swiftshader.dll	16/10/2024 16:13	Estensione dell'ap...	5.210 KB
vk_swiftshader_icd	16/10/2024 16:13	File di origine JSON	1 KB
vulkan-1.dll	16/10/2024 16:13	Estensione dell'an	907 KB

Comandi di installazione

Per avviare singolarmente i vari progetti in locale è bisogna effettuare la compilazione, utilizzando i seguenti comandi, che variano da servizio a servizio:

Servizio	Comando
User Service	mvn clean install
Exercise Service	mvn clean install
Compiler Service	npm install
Leaderboard Service	mvn clean install
Frontend Thin	npm install
Server-Setup	docker-compose up
Frontend Electron	npm install

Endpoint servizi

Di seguito sono elencati i vari endpoint per ogni servizio del software:

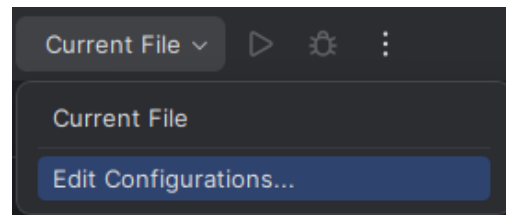
Servizio	Endpoint
User Service	http://localhost:8081
Exercise Service	http://localhost:9090
Compiler Service	http://localhost:8083
Leaderboard Service	http://localhost:8087
Frontend Thin	http://localhost:4200

Run Configurations

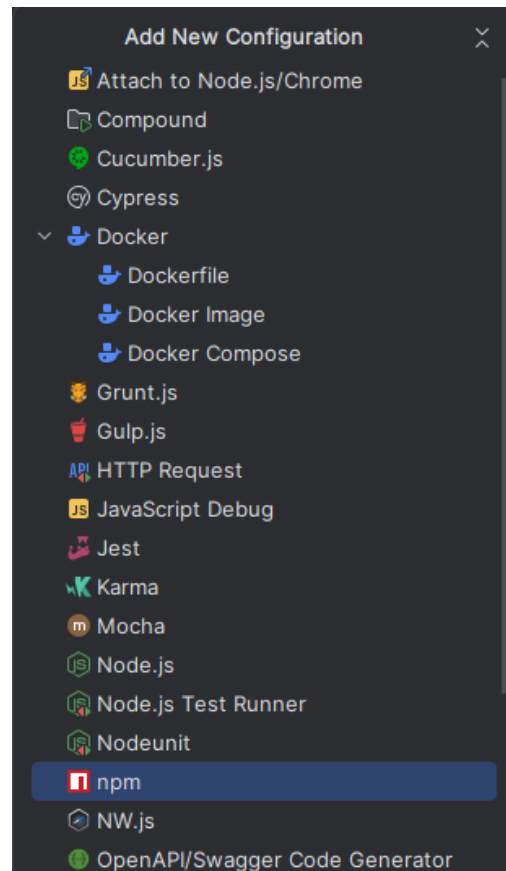
Per le seguenti configurazioni sono stati utilizzati **WebStorm** (npm e node.js) e **IntelliJ IDEA** (spring-boot run) come ambienti di sviluppo.

Frontend Thin Client

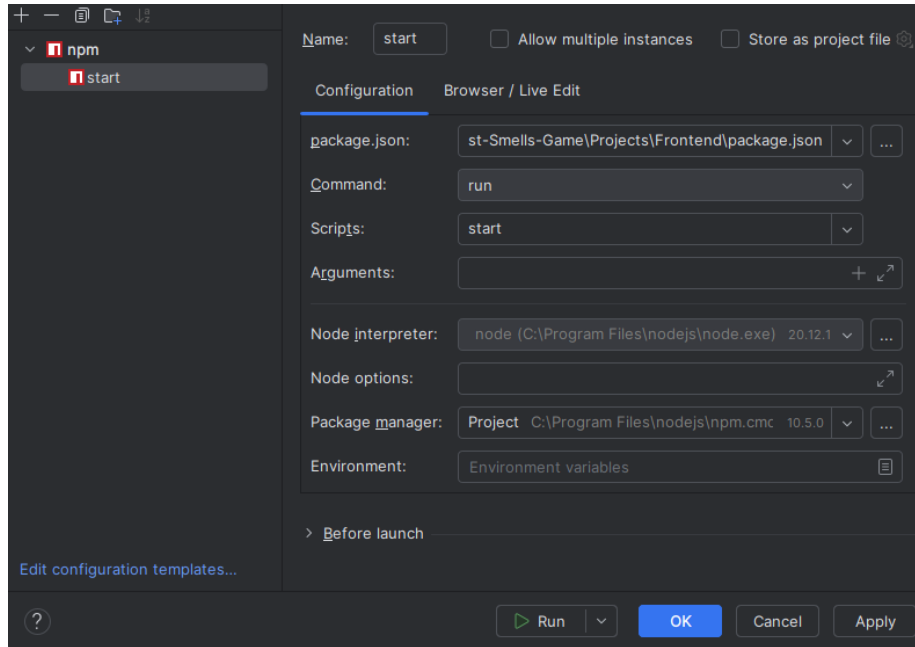
1. Crea una nuova configurazione cliccando su "Edit Configurations":



2. Clicca sul bottone "+" e tra la lista di configurazioni seleziona "npm":

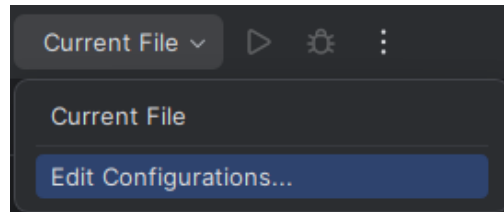


3. Inserisci i seguenti parametri nelle impostazioni:

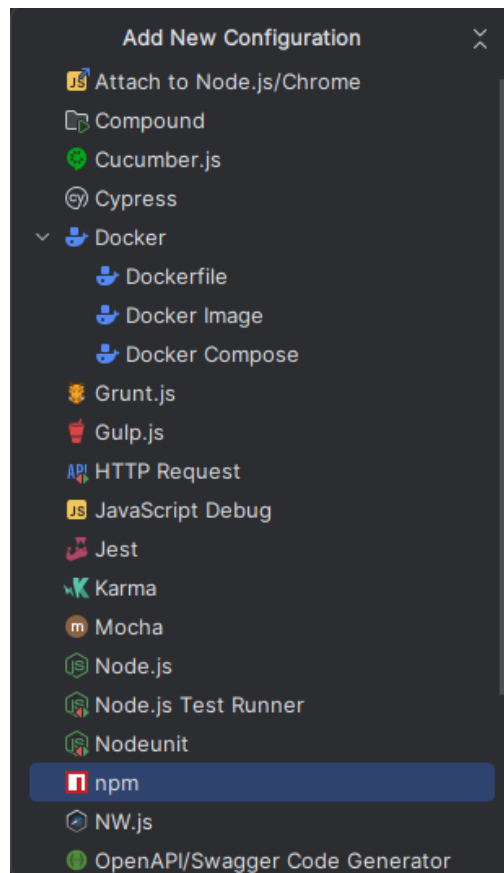


Frontend Electron

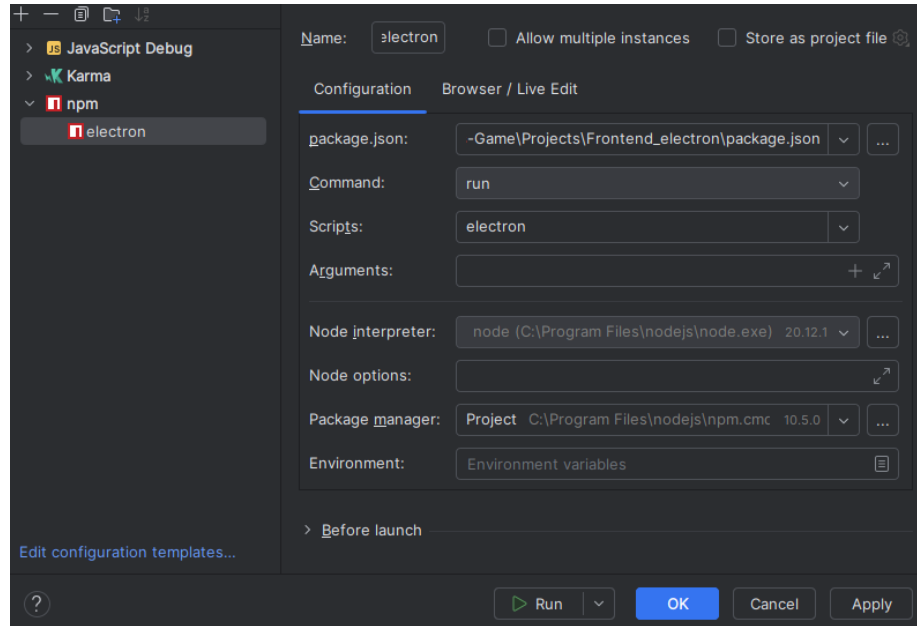
1. Crea una nuova configurazione cliccando su "Edit Configurations":



2. Clicca sul bottone "+" e tra la lista di configurazioni seleziona "npm":

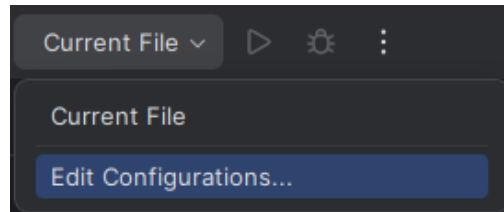


3. Inserisci i seguenti parametri nelle impostazioni:

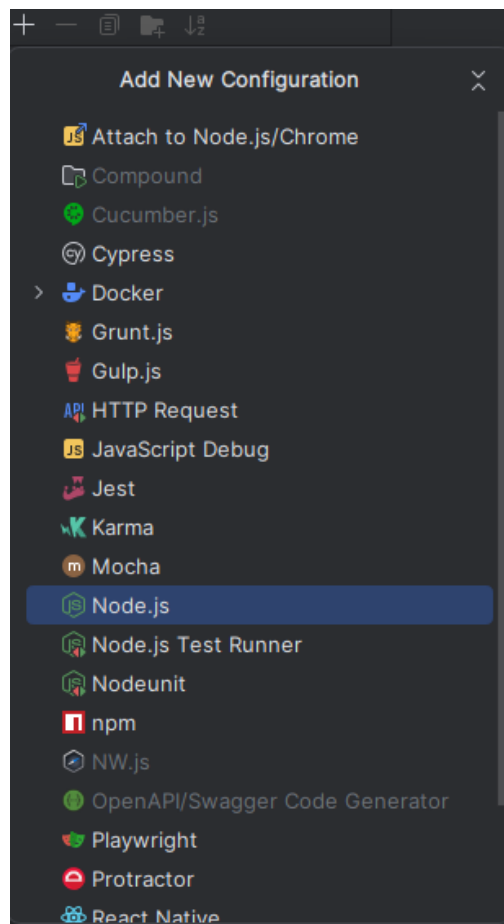


Compiler Service

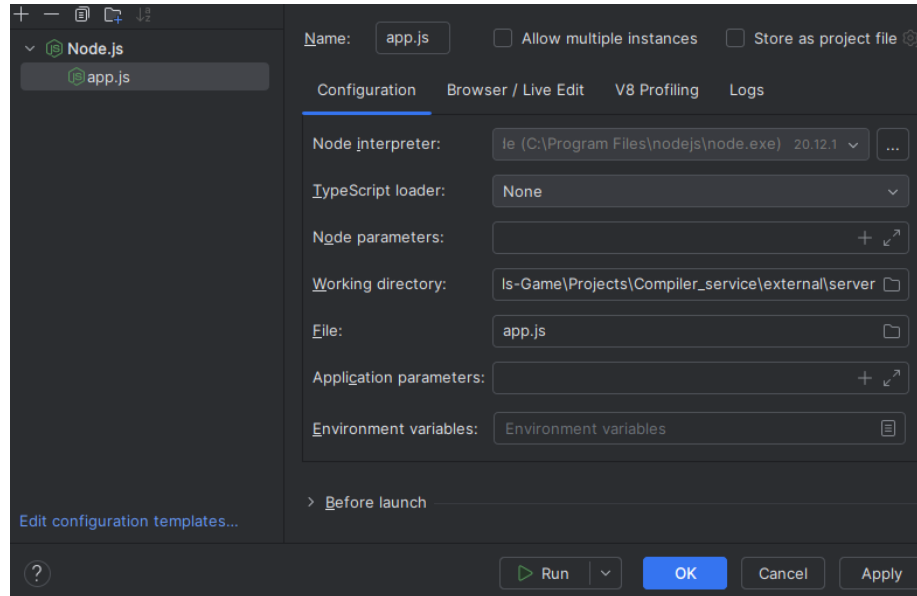
1. Crea una nuova configurazione cliccando su "Edit Configurations":



2. Clicca sul bottone "+" e tra la lista di configurazioni seleziona "Node.js":

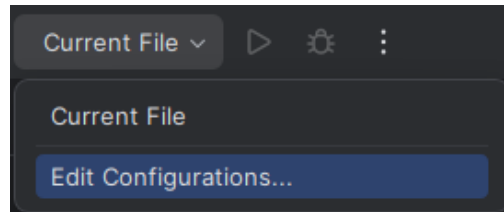


3. Inserisci i seguenti parametri nelle impostazioni:

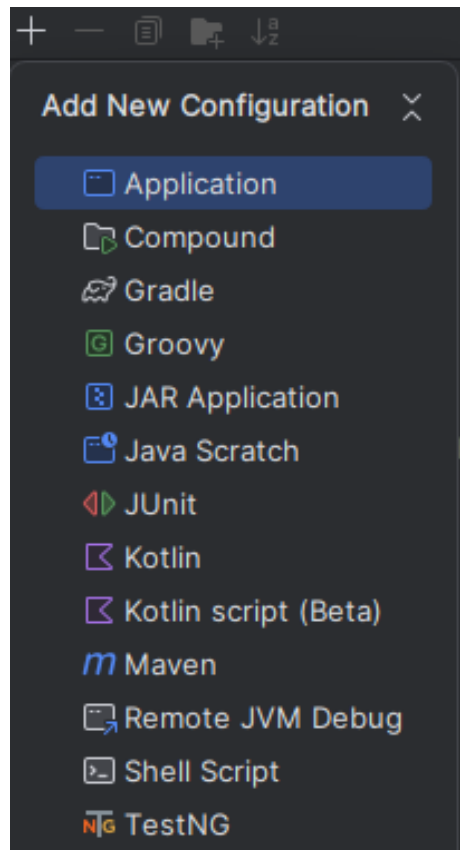


Exercise, Leaderboard & User Service

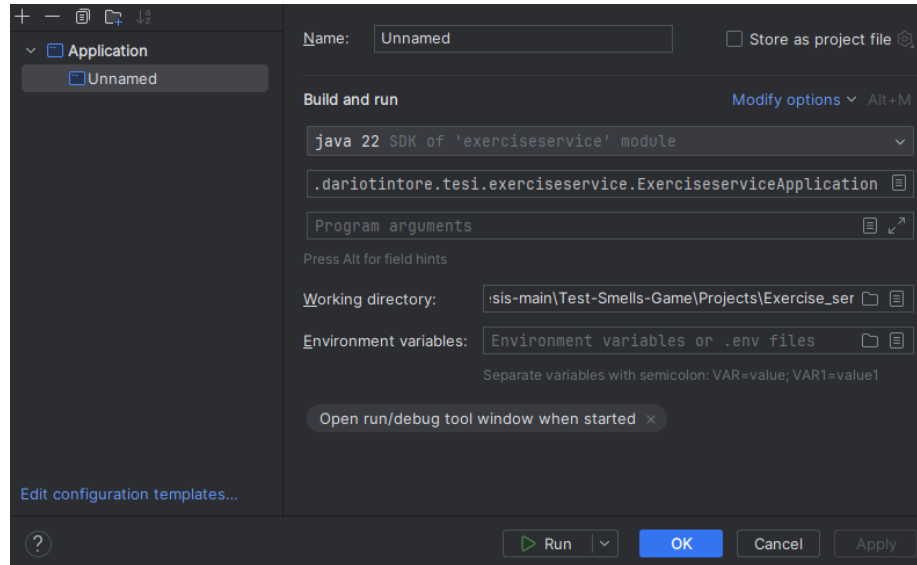
1. Crea una nuova configurazione cliccando su "Edit Configurations":



2. Clicca sul bottone "+" e tra la lista di configurazioni seleziona "Application":



3. Inserisci i seguenti parametri nelle impostazioni:



Anche per gli altri servizi Spring bisogna applicare la stessa procedura.

Configurazione Esercizio

I file di un determinato esercizio sono presenti nella directory "ExerciseDB/{NomeEsercizio}" dell'Exercise Service; qui troviamo 3 file:

- **{NomeEsercizio}.java**: classe dell'esercizio;
- **{NomeEsercizio}Test.java**: classe di test dell'esercizio;
- **{NomeEsercizio}Config.json**: file di configurazione.

La struttura di un file json di configurazione è la seguente:

```
{
  "refactoring_game_configuration": {
    "dependencies": [
      ""
    ],
    "refactoring_limit": number,
    "smells_allowed": number,
    "level": number,
    "ignored_smells": [
      ""
    ]
  },
  "check_game_configuration": {
    "questions": [
      {
        "questionTitle": "",
        "questionCode": "",
        "answers": [
          {
            "answerText": "",
            "isCorrect": boolean
          }
        ]
      }
    ]
  },
  "auto_valutative": boolean
}
```

Con questo file sarà possibile modificare i parametri di un esercizio, ovvero il livello, numero di smell ammessi (ad esempio, diminuendo tale valore rendiamo l'esercizio più difficile), le dipendenze necessarie, gli smell ignorati, il limite di copertura del codice e le domande e risposte per i quiz della modalità Check Game.

Configurazione Esperimento

Per configurare un esperimento secondo le proprie necessità, è possibile utilizzare i file json "esperimento1.json" e "esperimento2.json" che si troveranno nel volume docker dopo l'installazione del sistema, andando a modificarne i valori in maniera diretta accedendo al volume docker. Alternativamente è possibile creare nuovi file json ed aggiungerli manualmente all'interno del volume.

I file di configurazione di un esperimento hanno la seguente struttura:

```
{
  "name": string,
  "date": string (yyyy-mm-dd),
  "students": [
    {
      "name": string,
      "exercise": string,
      "start": string (hh:mm),
      "end": string (hh:mm),
      "submitted": boolean
    }
  ],
  "type": string (competitivo/collaborativo)
}
```

Configurazione Ngrok (Electron)

Di seguito sono elencate le istruzioni per ospitare una sessione di gioco utilizzando il client Electron e la versione gratuita di ngrok.

La configurazione consiste nell'esporre solo user service, leaderboard service ed exercise service (a causa del limite di tunnel della versione gratuita), ed utilizzare il Thick Client e il compiler locale (da impostare nella pagina settings).

In primo luogo, bisogna navigare nella directory "configurazione-ngrok" della repository del progetto, in cui sono presenti i file ngrok.exe (eseguibile scaricato dal sito ngrok) e ngrok.yml (file di configurazione della sessione ngrok). Il file di configurazione ha la seguente struttura:

```
version: "2"
authtoken: AUTH_TOKEN
tunnels:
  leaderboard_service:
    addr: 8087
    proto: http
  exercise_service:
    addr: 9090
    proto: http
  user_service:
    addr: 8081
    proto: http
```

Bisogna modificare tale file inserendo il proprio auth_token, al posto della stringa 'AUTH_TOKEN'. L'auth token, univoco per ogni utente, viene generato da ngrok ed è reperibile al seguente link: <https://dashboard.ngrok.com/get-started/your-authtoken>.

Il prossimo passaggio consiste nell'aprire un terminale (nella directory "configurazione-ngrok") ed eseguire il seguente comando, che avvia la sessione ngrok usando la configurazione del file yml, con tre tunnel per ciascun servizio esposto:

```
Thesis-main\configurazione-ngrok> ./ngrok start --all --config ngrok.yml
```

Dopo l'esecuzione del comando verrà avviata la procedura e nel terminare saranno presenti tutte le informazioni riguardo la sessione, tra cui i vari indirizzi pubblici di ogni servizio:

```
Session Status      online
Account             loriszannini.74@gmail.com (Plan: Free)
Version             3.18.0
Region              Europe (eu)
Latency             61ms
Web Interface       http://127.0.0.1:4040
Forwarding           https://5604-151-19-114-148.ngrok-free.app -> http://localhost:8081
Forwarding           https://b738-151-19-114-148.ngrok-free.app -> http://localhost:9090
Forwarding           https://d139-151-19-114-148.ngrok-free.app -> http://localhost:8087
```

Come ultimo passo, bisogna aprire il frontend Electron, dirigersi nella pagina delle impostazioni, impostare la modalità di compilazione in locale e settare gli indirizzi pubblici appena creati all'interno della pagina settings (il compiler resta invariato), per poter utilizzare i servizi in remoto:

The screenshot shows a settings dialog with two sections. The first section, 'Environments', contains five text input fields for different services, each with an 'OK' button below it. The second section, 'Compile mode', has two radio buttons: 'Local' (selected) and 'Cloud'. Below the 'Cloud' option is a red text message: 'To use cloud you need to login'.

Service	URL
User Service Url	https://5604-151-19-114-148.ngrok-free.app
Compiler Service url	http://localhost:8083
Exercise Service url	https://b738-151-19-114-148.ngrok-free.app
Leaderboard Service url	https://d139-151-19-114-148.ngrok-free.app

Compile mode

☒ Local

☐ Cloud

To use cloud you need to login

Per salvare tali modifiche bisogna cliccare il tasto 'OK'. A questo punto, sarà possibile utilizzare il software e tutte le sue funzionalità totalmente in remoto.