dgroove
Tecnologia emozionale.

ICT Training Center

Il tuo partner per la Formazione e la Trasformazione digitale della tua azienda

Spring AI

Generative Artificial Intelligence con Java

Simone Scannapieco

Corso avanzato per Venis S.p.A, Venezia, Italia

Novembre 2025

RAG Web Search

- Il *vector store* non è l'unico *data source* dal quale Spring AI può attingere documenti
  - `VectorStoreDocumentRetriever` unica implementazione di `DocumentRetriever` (Spring AI <= 1.0.3)
  - Possibile utilizzare informazione dal web (RAG *web search*)
    - Metodo efficace per mitigare la *knowledge cut-off* di LLM
    - ⚠ Utilizzo di connettori API appositi (es. Google Web Search API)

Interfaccia per recupero documenti

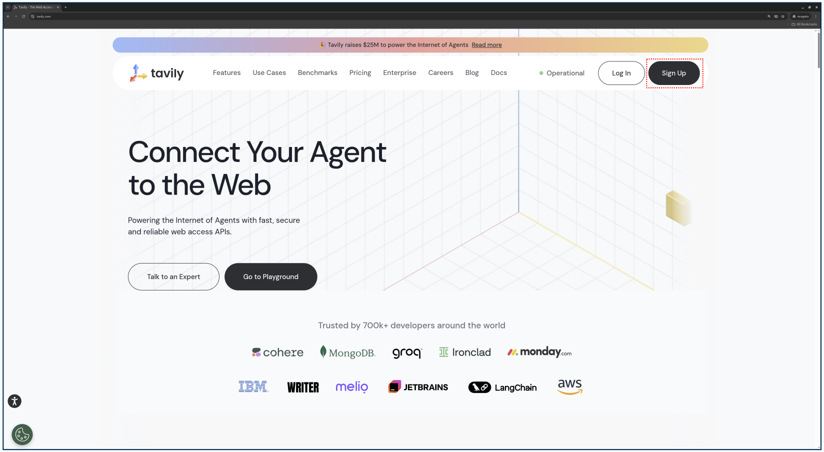
```java
/**
 * Component responsible for retrieving {@link Document}s from an underlying data source,
 * such as a search engine, a vector store, a database, or a knowledge graph.
 *
 * @author Christian Tzolov
 * @author Thomas Vitale
 * @since 1.0.0
 */
public interface DocumentRetriever extends Function<Query, List<Document>> {

        /**
         * Retrieves relevant documents from an underlying data source based on the given
         * query.
         * @param query The query to use for retrieving documents
         * @return The list of relevant documents
         */
        List<Document> retrieve(Query query);

        default List<Document> apply(Query query) {
                return retrieve(query);
        }

}
```

Progetto Spring AI
Applicazione e passaggi

- Servizio RAG Ollama Web Search
  1. Creazione *account* Tavily e API *key*
  2. Modifica variabili di ambiente
  3. Creazione implementazione *document retriever* tramite *web search*
  4. Modifica configurazione RAG
  5. Modifica interfaccia e implementazione servizio
  6. Modifica proprietà applicativo
  7. *Test* delle funzionalità con Postman/Insomnia

AMBIENTE DI SVILUPPO
CREAZIONE ACCOUNT TAVILY

1 Accedere al portale `https://tavily.com/` e cliccare il pulsante `Sign up`



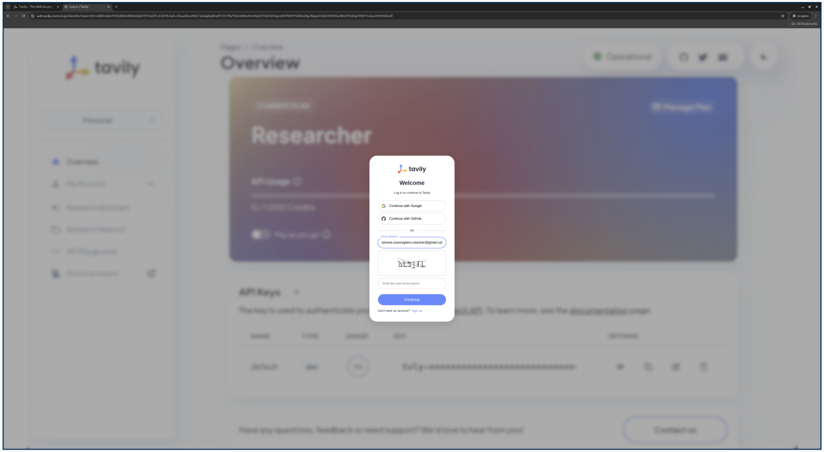
Connect Your Agent to the Web

Powering the Internet of Agents with fast, secure and reliable web access APIs.

Trusted by 700k+ developers around the world

Ambiente di sviluppo
Creazione account Tavily

2 Creare una nuova utenza


Welcome

Log in to continue to Tavily

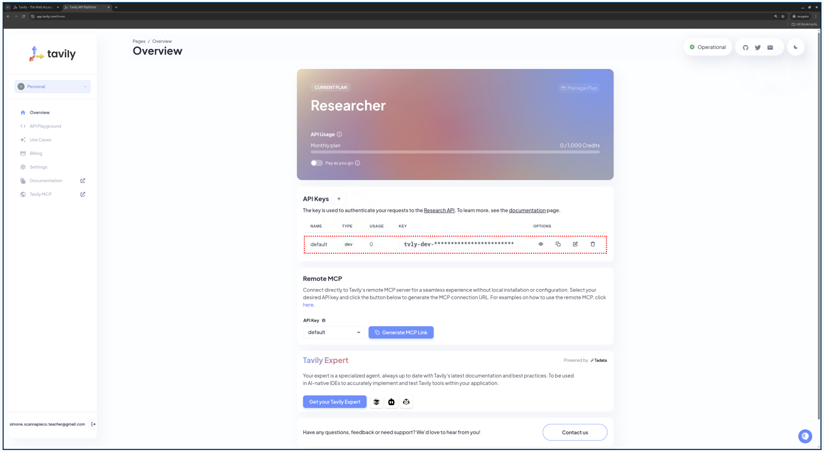
Email address*: simone.scannapieco.teacher@gmail.com

Enter the code shown above*

Don't have an account? Sign up

Ambiente di sviluppo
Creazione account Tavily

3 Accedere al portale e selezionare la API *key* di *default*



Pages / Overview

Overview

CURRENT PLAN

Researcher

API Usage

Monthly plan 0 / 1,000 Credits

Pay as you go

API Keys

The key is used to authenticate your requests to the Research API. To learn more, see the documentation page.

NAME	TYPE	USAGE	KEY	OPTIONS
default	dev	0	tvly-dev-************************	

Remote MCP

Connect directly to Tavily's remote MCP server for a seamless experience without local installation or configuration. Select your desired API key and click the button below to generate the MCP connection URL. For examples on how to use the remote MCP, click here.

API Key: default

Tavily Expert

Your expert is a specialized agent, always up to date with Tavily's latest documentation and best practices. To be used in AI-native IDEs to accurately implement and test Tavily tools within your application.

Have any questions, feedback or need support? We'd love to hear from you!

simone.scannapieco.teacher@gmail.com

Progetto Spring AI
Variabili di ambiente - Applicativo Spring

File `launch.json`

```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "type": "java",
            "name": "Current File",
            "request": "launch",
            "mainClass": "${file}"
        },
        {
            "type": "java",
            "name": "DemoApplication",
            "request": "launch",
            "mainClass": "it.venis.ai.spring.demo.DemoApplication",
            "projectName": "demo",
            "env": {
                "GOOGLE_AI_API_KEY": "...",
                "HUGGING_FACE_HUB_TOKEN": "...",
                "TAVILY_SEARCH_API_KEY": "..."
            }
        }
    ]
}
```

Progetto Spring AI
Variabili di ambiente - JUnit test

File `settings.json`

```
{
    "java.configuration.updateBuildConfiguration": "interactive",
    "java.test.config": {
        "env": {
            "GOOGLE_AI_API_KEY": "...",
            "HUGGING_FACE_HUB_TOKEN": "...",
            "TAVILY_SEARCH_API_KEY": "..."
        }
    }
}
```

Document retriever via web search - I

```java
package it.venis.ai.spring.demo.rag;

...

public class WebSearchDocumentRetriever implements DocumentRetriever {

    private static final Logger logger = LoggerFactory.getLogger(WebSearchDocumentRetriever.class);

    private static final String TAVILY_API_KEY = "TAVILY_SEARCH_API_KEY";
    private static final String TAVILY_BASE_URL = "https://api.tavily.com/search";
    private static final int DEFAULT_RESULT_LIMIT = 5;
    private final int resultLimit;
    private final RestClient restClient;

    public WebSearchDocumentRetriever(RestClient.Builder clientBuilder, int resultLimit) {
        Assert.notNull(clientBuilder, "clientBuilder cannot be null");
        String apiKey = System.getenv(TAVILY_API_KEY);
        Assert.hasText(apiKey, "Environment variable " + TAVILY_API_KEY + " must be set");
        this.restClient = clientBuilder
                .baseUrl(TAVILY_BASE_URL)
                .defaultHeader(HttpHeaders.AUTHORIZATION, "Bearer " + apiKey)
                .build();
        if (resultLimit <= 0) {
            throw new IllegalArgumentException("resultLimit must be greater than 0");
        }
        this.resultLimit = resultLimit;
    }

    ...
```

Document retriever via *web search* - II

```java
...
@Override
public List<Document> retrieve(Query query) {
    Assert.notNull(query, "query cannot be null");
    String q = query.text();
    Assert.hasText(q, "query.text() cannot be empty");

    TavilyResponsePayload response = restClient.post()
            .body(new TavilyRequestPayload(q, "advanced", resultLimit))
            .retrieve()
            .body(TavilyResponsePayload.class);

    if (response == null || CollectionUtils.isEmpty(response.results())) {
        return List.of();
    }

    List<Document> docs = new ArrayList<>(response.results().size());
    for (TavilyResponsePayload.Hit hit : response.results()) {
        // Map each Tavily hit into a Spring AI Document with metadata and score.
        Document doc = Document.builder()
                .text(hit.content())
                .metadata("title", hit.title())
                .metadata("url", hit.url())
                .score(hit.score())
                .build();
        docs.add(doc);
    }
    return docs;
}

@JsonNaming(PropertyNamingStrategies.SnakeCaseStrategy.class)
record TavilyRequestPayload(String query, String searchDepth, int maxResults) {}
...
```

Document retriever via *web search* - III

```java
    ...

    record TavilyResponsePayload(List<Hit> results) {
        record Hit(String title, String url, String content, Double score) {}
    }

    public static Builder builder() {
        return new Builder();
    }

    public static class Builder {
        private RestClient.Builder clientBuilder;
        private int resultLimit = DEFAULT_RESULT_LIMIT;

        private Builder() {}

        public Builder restClientBuilder(RestClient.Builder clientBuilder) {
            this.clientBuilder = clientBuilder;
            return this;
        }

        public Builder maxResults(int maxResults) {
            if (maxResults <= 0) {
                throw new IllegalArgumentException("maxResults must be greater than 0");
            }
            this.resultLimit = maxResults;
            return this;
        }

        public WebSearchDocumentRetriever build() {
            return new WebSearchDocumentRetriever(clientBuilder, resultLimit);
        }
    }
}
```

Configurazione Vector Store

```java
package it.venis.ai.spring.demo.config;

import it.venis.ai.spring.demo.rag.WebSearchDocumentRetriever;

...

@Configuration
public class RAGConfig {

    ...

    @Bean
    public RetrievalAugmentationAdvisor webSearchRetrievalAugmentationAdvisor(RestClient.Builder restClientBuilder) {

        return RetrievalAugmentationAdvisor.builder()
                .documentRetriever(WebSearchDocumentRetriever.builder()
                        .restClientBuilder(restClientBuilder).maxResults(1).build())
                .build();
    }

}
```

Interfaccia servizio

```
package it.venis.ai.spring.demo.services;

import it.venis.ai.spring.demo.model.Answer;
import it.venis.ai.spring.demo.model.QuestionRequest;

public interface RAGService {

    public Answer getGeminiRAGAnswer(QuestionRequest request);

    public Answer getOllamaRAGAnswer(QuestionRequest request);

    public Answer getOllamaWebSearchRAGAnswer(QuestionRequest request);

}
```

Implementazione servizio

```
package it.venis.ai.spring.demo.services;

...

@Service
@Configuration
public class RAGServiceImpl implements RAGService {

    ...
    private RetrievalAugmentationAdvisor webSearchRetrievalAugmentationAdvisor;

    public RAGServiceImpl(
        @Qualifier("geminiChatClient") ChatClient geminiChatClient,
        @Qualifier("ollamaChatClient") ChatClient ollamaChatClient,
        @Qualifier("ollamaMemoryChatClient") ChatClient ollamaMemoryChatClient,
        @Qualifier("geminiVectorStore") VectorStore geminiVectorStore,
        @Qualifier("ollamaVectorStore") VectorStore ollamaVectorStore,
        @Qualifier("geminiRetrievalAugmentationAdvisor") RetrievalAugmentationAdvisor geminiRetrievalAugmentationAdvisor,
        @Qualifier("webSearchRetrievalAugmentationAdvisor") RetrievalAugmentationAdvisor webSearchRetrievalAugmentationAdvisor)

        this.geminiChatClient = geminiChatClient;
        this.ollamaChatClient = ollamaChatClient;
        this.ollamaMemoryChatClient = ollamaMemoryChatClient;
        this.geminiVectorStore = geminiVectorStore;
        this.ollamaVectorStore = ollamaVectorStore;
        this.geminiRetrievalAugmentationAdvisor = geminiRetrievalAugmentationAdvisor;
        this.webSearchRetrievalAugmentationAdvisor = webSearchRetrievalAugmentationAdvisor;
    }

    ...
```

Implementazione controllore REST

```java
package it.venis.ai.spring.demo.controllers;

...

@RestController
public class QuestionController {

    ...

    @PostMapping("/ollama/ask/rag")
    public Answer getOllamaRAGAnswer(@RequestBody QuestionRequest request) {

        return this.ragService.getOllamaRAGAnswer(request);

    }

    @PostMapping("/ollama/ask/rag/web-search")
    public Answer getOllamaWebSearchRAGAnswer(@RequestBody QuestionRequest request) {

        return this.ragService.getOllamaWebSearchRAGAnswer(request);

    }

}
```

Codice
Branch di riferimento

`https://github.com/simonescannapieco/spring-ai-advanced-dgroove-venis-code.git`

Branch: `11-spring-ai-gemini-ollama-rag-web-search`