

# Guida completa: usare Agent Skills con GitHub Copilot

## (VS Code e IntelliJ) + esempi reali Spring Java e Angular

---

### Cos'è una Agent Skill e perché dovresti usarla

Una Agent Skill è un modulo strutturato che estende il comportamento del coding agent di GitHub Copilot.

Non è un semplice prompt salvato.

È una cartella con un file SKILL.md (più eventuali script o template) che contiene:

- regole operative
- linee guida
- standard architetturali
- formato dell'output richiesto
- checklist

Quando usi Copilot in Agent Mode, l'agente:

1. legge il tuo prompt
2. verifica se esiste una Skill coerente
3. carica automaticamente le istruzioni della Skill
4. esegue il task seguendo quelle regole

Risultato:

l'AI non lavora più in modo generico, ma secondo pattern ripetibili e coerenti.

---

## Dove trovare Skills già pronte

Puoi:

- creare le tue Skill personalizzate
- oppure installare Skill già pronte da marketplace come skills.sh

***<https://skills.sh/>***

skills.sh è un directory di repository contenenti Skill compatibili con lo standard Agent Skills.

Esempi utili per uno stack Angular + Spring:

### Angular

- angular-architect
- angular-best-practices
- debug:angular

### Java

- best-java-structure
- lombok-patterns

Nota importante: molte Skill su skills.sh non sono “ufficiali GitHub”, ma compatibili con lo standard Agent Skills.

---

## Installare una Skill da skills.sh

Ogni Skill fornisce un comando tipo:

***`npx skills add https://github.com/<owner>/<repo> --skill <skill-name>`***

Esempio:

***npm skills add https://github.com/remotion-dev/skills --skill remotion-best-practices***

L'installazione scarica la cartella Skill.

Ma il punto cruciale è questo:

---

## **4 Dove posizionare le Skills per GitHub Copilot**

GitHub Copilot cerca le Agent Skills in percorsi specifici.

### **Opzione A — Nel repository (consigliata per team)**

Struttura:

```
<repo>/  
  .github/  
    skills/  
      spring-java/  
        SKILL.md  
      angular/  
        SKILL.md
```

Vantaggi:

- versionate su Git
- condivise col team
- stesso comportamento per tutti

---

### **Opzione B — Locali (solo per te)**

Posizione tipica:

```
~/copilot/skills/
```

Oppure struttura equivalente per skill utente.

---

# Usare le Skills in VS Code (percorso più stabile oggi)

## Passaggi

1. Installa GitHub Copilot e Copilot Chat
2. Abilita Agent Mode
3. Assicurati che le Skill siano in .github/skills/
4. Apri Copilot Chat → Agent Mode
5. Assegna un task multi-file

## Esempio Spring

Prompt:

## Esempio Angular

Prompt:

Copilot caricherà automaticamente la Skill se la descrizione è coerente con il task.

---

# Usare le Skills in IntelliJ IDEA (JetBrains)

Copilot Agent Mode in JetBrains è attualmente in public preview.

## Passaggi

1. Aggiorna IntelliJ
2. Installa/aggiorna plugin GitHub Copilot

3. Login con account Copilot attivo
4. Abilita Copilot Chat e Agent Mode
5. Inserisci le Skill nel repo `.github/skills/`
6. Apri Copilot Chat → Agent Mode
7. Assegna task multi-file

Nota: la discovery delle Skill può essere più stabile su VS Code rispetto a IntelliJ finché la preview non è completamente consolidata.

---

## Esempio reale di SKILL.md — Spring Java

Percorso:

`.github/skills/spring-java/SKILL.md`

Contenuto:

---

***name: spring-java***

***description: Standard Spring Boot per controller, service, DTO, error handling e test.***

---

***Regole:***

- ***Controller sottili, logica in service***
- ***DTO per boundary API***
- ***@Valid per validazione***
- ***ControllerAdvice per gestione errori***
- ***Logging strutturato***
- ***Test: unit su service, slice su controller***

***Output richiesto:***

1. ***Piano di refactor***
  2. ***Patch multi-file***
  3. ***Checklist test e regressione***
- 

## 8 Esempio reale di SKILL.md — Angular

`.github/skills/angular/SKILL.md`

---

**name:** *angular*

**description:** *Best practice Angular enterprise*

---

**Regole:**

- *Strict typing, no any*
- *RxJS pulito, no nested subscribe*
- *OnPush dove possibile*
- *Lazy loading moduli*
- *Struttura feature-based*

**Output:**

- *Refactor proposto*
- *Diff suggerito*
- *Miglioramenti performance*
- *Strategia test*

---

## Riferimenti ufficiali

### GitHub – Agent Skills

<https://docs.github.com/en/copilot/concepts/agents/about-agent-skills>

### GitHub Changelog – Copilot supports Agent Skills

<https://github.blog/changelog/2025-12-18-github-copilot-now-supports-agent-skills/>

### GitHub Changelog – Agent Mode in JetBrains

<https://github.blog/changelog/2025-05-19-agent-mode-and-mcp-support-for-copilot-in-jetbrains-eclipse-and-xcode-now-in-public-preview/>

### Skills pronte e pubbliche – skills.sh

<https://skills.sh/>