

### ◆ Step 1: Hooks hotay kya hain?

☞ **system ke andar ek interception point dena** — jahan tum code inject karke system ka default behavior modify ya extend kar sakti ho.

Hooks ka matlab hota hai "**ek aisi jagah jahan tum apna code daal sakti ho, taake koi cheez automatic run ho jaye kisi special waqt pe**".

☞ Simple lafzon main:

- Hooks = **choti choti points** jahan tum system ko bol sakti ho "yeh kaam is waqt karna"
- 

### ◆ Step 2: Hooks kyun use karte hain?

Hum hooks is liye use karte hain taake:

1. **Apna extra code** kisi process ke start ya end main daal saken.
  2. System ke behavior ko **customize** kar saken.
  3. Har jagah code repeat na karna pare.
- 

### ◆ Step 3: Example (daily life)

Maan lo tum school ja rahi ho 🎒.

- Gate pe guard check karta hai = ek **hook** (process start hone se pehle check).
- Class khatam hoti hai aur bell bajti hai 🛎 = dusra **hook** (process end hone ke baad action).

Yani, system ke andar kuch **points fix hote hain** jahan tum apni "extra activity" chipka sakti ho.

---

### ◆ Step 4: Programming world main Hooks

Programming (aur AI Agentic SDK) main hooks ka kaam hai:

- Agent jab **start ho** → tum apna code chala sakti ho.
- Agent jab **kisi tool ko call kare** → tum us waqt kuch log ya check kar sakti ho.
- Agent jab **reply bhejne lage** → tum us reply ko edit/change kar sakti ho.

## ◆ Types of Hooks (advance samajh ke live)

### 1. Lifecycle Hooks

- Ye system ke lifecycle (shuru → beech → end) ke dauran chalti hain.
- Example:
  - Agent jab start ho → `on_start`
  - Agent jab reply complete kare → `on_result`

### 2. Event Hooks

- Jab koi **special event** hota hai, tab ye trigger hoti hain.
- Example:
  - Tool call hua → `on_tool_call`
  - Error aayi → `on_error`

### 3. Custom Hooks

- Jo tum khud banati ho apne project ke liye.
  - Example: "Jab bhi agent message banaye, usme emoji add karna 😊".
- 

## ◆ Agentic AI SDK main Hooks ka Flow

- `on_start` → Agent start hua
- `on_end` → Agent end hua (final result ke sath)
- `on_tool_start` → Tool call start hua
- `on_tool_end` → Tool call end hua
- `on_handoff` → Jab ek agent dusre agent ko handoff karta hai

## ◆ 1. Agent Hooks (chhoti soch / local scope)

- Ye sirf ek **specific agent** ke liye kaam karte hain.
- Agar tumhari app me sirf **ek agent** chal raha hai (jaise Auntie Rishta Agent 🧓), to **Agent hooks** best hain, simple aur easy.

## ◆ Runner Hooks (bari soch / global scope)

- Runner ek **controller** hota hai jo ek ya zyada agents ko chalata hai.
- Agar tumhari app me **multiple agents** hain (jaise: Auntie agent + Filtering agent + Messaging agent), aur tum chah rahi ho ke har ek ka start/end ek jagah log ho, to **Runner hooks** use karte hain.

### ◆ 3. Kaunsa zyada ahm hai?

- Agar tum sirf **ek agent bana rahi ho** → **Agent hooks** kaafi hain ✓
- Agar tum **multi-agent system** bana rahi ho (zyada complex AI apps) → **Runner hooks** zaroori hote hain ✓

### ◆ Easy Answer (exam/viva style):

☞ “Agent hooks ek agent ke lifecycle ko control karte hain. Runner hooks poore system ya multiple agents ke lifecycle ko monitor karte hain. Chhoti apps ke liye agent hooks use hote hain, aur bari complex apps ke liye runner hooks ahm hote hain.”

#### 1. Lifecycle Hooks:

- Hooks asal main woh jagah hote hain jahan tum **apna khud ka code** chala sakti ho.
- Yeh agent ke process ke **khaas moments** pe trigger hote hain.
  - Jaise: jab agent start hota hai, tool call karta hai, ya end hota hai.
- Matlab tum apni **extra logic** inject kar sakti ho bina agent ka asal code badle.

#### 2. Model Settings:

- Model settings ka matlab hai ke tum AI ke **behavior** ko control kar sakti ho.
- Example:
  - Temperature change karke model ko **zyada creative ya zyada accurate** banaya ja sakta hai.
  - Max tokens decide karte hain ke **kitna lamba jawab** aaye.
- In settings se tum model ka **tone, detail aur style** adjust karti ho.

---

Do qisam ke hooks hote hain:

#### **1. Agent Hooks**

- Ye sirf ek **individual agent** par focus karte hain.
- Ye tumhe allow karte hain ke tum apni **custom logic** agent ke workflow ke andar inject kar sako.
- In se tum track kar sakti ho:
  - Kab agent start hua (processing shuru ki)
  - Kab agent ne apna kaam complete kiya
  - Kab agent ne kisi **external tool** ke sath interaction kiya

## Run Hooks

- Ye aise hooks hote hain jo **global events** ko manage karte hain.
  - Matlab yeh sirf ek agent tak limited nahi hote, balki poore **execution (run)** ko cover karte hain, chahe ek agent ho ya ek se zyada agents.
  - In ki madad se tum yeh monitor aur control kar sakti ho:
    - Agent ka **start** aur **end** (execution kab shuru hui, kab khatam hui)
    - Jab koi **tool call** hota hai
    - Jab agents ke darmiyan **handoff** (control pass) hota hai
- 

## 🔧 Model Settings: Apne AI Agent ke Dimaag ko Control Karna

Aap AI model ke settings ko adjust karke uske responses ko control kar sakte ho. Is se aap uski **personality (shakhsiyat)** aur **performance (kaam karne ka tareeqa)** ko apni zarurat ke mutabiq badal sakte ho.

◆ **Temperature:** Ye decide karta hai ke agent kitna **creative (naya sochne wala)** ya **focused (seedha jawab dene wala)** hoga.

◆ **Tool Choice:** Aap set karte ho ke agent kaunse tools use kar sakta hai, jaise **calculator, weather app**, etc.

◆ **Max Tokens:** Ye control karta hai ke agent ka jawab **kitna lamba ya chhota** ho sakta hai.

◆ **Parallel Tools:** Ye option decide karta hai ke agent ek waqt me **multiple tools (zyada tools)** use kar sakta hai ya sirf ek hi.

📖 Matlab simple lafzon me, **Model Settings** aapko **full control** deti hain ke **AI agent** kaise soche, kaise behave kare aur kitna **detailed jawab** de.

---

---