Gelato Automate SDK

Create automated task executions using our typescript SDK:

NPM Package

?

How to use

Installautomate-sdk :

```

Copy yarn add @gelatonetwork/automate-sdk

```

ImportAutomateSDK :

```

Copy import{ AutomateSDK }from"@gelatonetwork/automate-sdk";

```

Instantiate the SDK with your signer:

```

Copy constautomate=newAutomateSDK(chainId,signer);

```

UsecreateTask to automate your function calls:

```

Copy interfaceCreateTaskOptions{ name:string;// your task name

// Function to execute execAddress:string;// address of your target smart contract execSelector:string;// function selector to execute on your target smart contract execAbi?:string;// ABI of your target smart contract

// Proxy caller dedicatedMsgSender:boolean;// task will be called via a dedicated msg.sender which you can whitelist (recommended: true)

// Optional: Pre-defined / static target smart contract inputs execData?:string;// exec call data

// Optional: Dynamic target smart contract inputs (using a resolver) resolverAddress?:string;// resolver contract address resolverData?:string;// resolver call data (encoded data with function selector) resolverAbi?:string;// your resolver contract ABI

// Optional: Time based task params interval?:number;// execution interval in seconds startTime?:number;// start timestamp in seconds or 0 to start immediately (default: 0)

// Optional: Single execution task singleExec?:boolean;// task cancels itself after 1 execution if true.

// Optional: Payment params useTreasury?:boolean;// use false if your task is self-paying (default: true) }

constparams:CreateTaskOptions={ name,execAddress,execSelector,interval };
const{taskId,tx}:TaskTransaction=awaitautomate.createTask(params);

```

Examples

Deploy a contract & automate your function call:

```

Copy // Deploying Counter contract constcounterFactory=awaithre.ethers.getContractFactory("Counter");
constcounter=awaitcounterFactory.deploy(GELATO_ADDRESSES[chainId].automate); awaitcounter.deployed();
```

// Call Counter.increaseCount(42) every 10 minutes const{taskId,tx}:TaskTransaction=awaitautomate.createTask({ execAddress:counter.address, execSelector:counter.interface.getSighash("increaseCount(uint256)"), execData:counter.interface.encodeFunctionData("increaseCount",[42]), execAbi:counter.interface.format("json")asstring, interval:10*60,// execute every 10 minutes name:"Automated counter every 10min", dedicatedMsgSender:true });

```

Use a resolver contract to automate your function call:

If you need more configurable execution condition and/or dynamic input data, you can create a task using a resolver function (learn how to write a resolver ).

```

Copy // Prepare Task data to automate constcounter=newContract(COUNTER_ADDRESSES,counterAbi,signer); constresolver=newContract(COUNTER_RESOLVER_ADDRESSES,counterResolverAbi,signer); constselector=counter.interface.getSighash("increaseCount(uint256)"); constresolverData=resolver.interface.getSighash("checker()");

// Create task const{taskId,tx}:TaskTransaction=awaitautomate.createTask({ execAddress:counter.address, execSelector:selector, resolverAddress:resolver.address, resolverData:resolverData, name:"Automated counter using resolver", dedicatedMsgSender:true });

```

Enable dedicated msg.sender:

To have a custommsg.sender that you can whitelist on your contract, you can enable thededicatedMsgSender flag.

```

Copy // Prepare Task data to automate constcounter=newContract(COUNTER_ADDRESSES,counterAbi,signer); constresolver=newContract(COUNTER_RESOLVER_ADDRESSES,counterResolverAbi,signer); constselector=counter.interface.getSighash("increaseCount(uint256)"); constresolverData=resolver.interface.getSighash("checker()");

// Create task const{taskId,tx}:TaskTransaction=awaitautomate.createTask({ execAddress:counter.address, execSelector:selector, resolverAddress:resolver.address, resolverData:resolverData, dedicatedMsgSender:true, name:"Automated counter using resolver", });

// Get dedicated proxy address to whitelist const{address,isDeployed}=awaitautomate.getDedicatedMsgSender()

```

More examples in our Hello World repository:

?

Previous Gelato Automate UI Next Smart Contract Last updated3 months ago On this page *NPM Package * How to use * Examples
```