Codecs and Encryption

Copy for LLM View Markdown

The Temporal Service persists data from your Workflow Executions, including inputs, outputs, and results. To protect sensitive data, use a Payload Codec to encrypt payloads before they reach the Temporal Service. With encryption enabled, data exists unencrypted only on the Client and the Worker process, on hosts that you control.

Alternatively, use Temporal Proxy to encrypt payloads before they reach the Temporal Service without changing Worker or Client code. You run the proxy in your environment, where it has access to plaintext payloads and your KMS. See the proxy's encryption guide for configuration and coverage limits.

The following data is persisted in the Event History and can be encrypted:

- Inputs and outputs/results in your Workflow, Activity, and Child Workflow
- Signal inputs
- Memo
- Headers (verify if applicable to your SDK)
- Query inputs and results
- Results of Local Activities and Side Effects
- Application errors and failures. Failure messages and call stacks are not encoded as codec-capable Payloads by default; you must explicitly enable encoding these common attributes on failures. For more details, see Failure Converter.

To v
sect

For

- TypeScript

Feedback

COOKIE POLICY

We use necessary cookies to make our site work. With your permission, we also use optional cookies to analyze site usage and assist in our marketing efforts. Read more in our cookie policy.

Do not sell or share my personal information.

Accept Decline

- .NET

Codec Server setup

Use a Codec Server to programmatically decode your encoded payloads.

A Codec Server is an HTTP server that uses your custom Codec logic to decode your data remotely. The Codec Server is independent of the Temporal Service and decodes your encrypted payloads through predefined endpoints. You create, operate, and manage access to your Codec Server in your own environment. When you configure a Codec Server endpoint in the Web UI or CLI, the Web UI and CLI use the remote endpoint to send and receive payloads from the Codec Server. See API contract requirements.

Decoded payloads can then be displayed in the Workflow Execution Event History on the Web UI. When you use a Codec Server, the decoded payloads are decoded and returned on the client side only. Payloads on the Temporal Service (whether on Temporal Cloud or self-hosted) remain encrypted.

Because you create, operate, and manage access to your Codec Server in your controlled environment, ensure that you consider the following:

- When you register a Codec Server endpoint with your Web UI, expect the Codec Server to receive multiple requests per Workflow Execution.
- Ensure that you secure access to your Codec Server. For details, see Authorization. You might need some form of Key management infrastructure for sharing your encryption keys between the Workers and your Codec Server.
-
-

COOKIE POLICY

We use necessary cookies to make our site work. With your permission, we also use optional cookies to analyze site usage and assist in our marketing efforts. Read more in our cookie policy.

Do not sell or share my personal information.

Your
your

API

When you create your Codec Server to handle requests from the Web UI, the following requirements must be met.

Endpoints

The Web UI and CLI send POST requests to the following endpoints on your Codec Server:

- `/decode` passes incoming payloads to the decode method in your Payload Codec.
- `/encode` passes incoming payloads to the encode method in your Payload Codec.
- `/download` retrieves and decodes payloads from External Storage. This endpoint is only needed if your Workers use External Storage. See Codec Server with External Storage for details.

For examples on how to create your Codec Server, see the following Codec Server implementation samples:

- Go
- Java
- Python
- TypeScript
- .NET

You can also add a verification step to check whether the incoming request has the required authorization to access the decode logic in your Payload Codec.

Headers

Each

COOKIE POLICY

We use necessary cookies to make our site work. With your permission, we also use optional cookies to analyze site usage and assist in our marketing efforts. Read more in our cookie policy.

Do not sell or share my personal information.

For details on setting up authorization, see Authorization.

Request body

The general specification for the `POST` request body contains payloads. By default, all field values in your payload are base64 encoded, regardless of whether they are encrypted by your custom codec implementation.

The following example shows a sample `POST` request body with base64 encoding.

```
{
  "payloads": [{
    "metadata": {
      "encoding": <base64EncodedEncodingHint>
    },
    "data": <encryptedPayloadData>
  }, ...]
}
```

CORS

By default, in cross-origin Fetch/XHR invocations, browsers will not send credentials. Enable Cross-Origin Resource Sharing (CORS) requests on your Codec Server to receive HTTP/HTTPS requests from the Temporal Web UI.

At a minimum, enable the following responses from your Codec Server to allow requests coming from the Temporal Web UI:

- `Access-Control-Allow-Origin`
-
-

For
follo

COOKIE POLICY

We use necessary cookies to make our site work. With your permission, we also use optional cookies to analyze site usage and assist in our marketing efforts. Read more in our cookie policy.

Do not sell or share my personal information.

For details on what a sample request/response looks like from the Temporal Web UI, see Sample Request/Response. If setting authorization, include `Authorization` in your `Access-Control-Allow-Headers`. For details on setting up authorization, see Authorization.

Authorization

It is important to establish how you will provide access to your Codec Server. Because it is designed to decode potentially sensitive data with a single API call, access to a production Codec Server should be restricted.

Depending on your infrastructure and risk levels, it might be sufficient to restrict HTTP ingress to your Codec Server (such as by using a VPN like WireGuard). The Temporal Web UI can communicate with a Codec Server that is only accessible on `localhost`, so this is a legitimate security pattern. However, if your Codec Server is exposed to the internet at all, you will likely need an authentication solution.

If you are already using an organization-wide authentication provider, you should integrate it with your Codec Server. Remember, a Codec Server is just a standalone HTTP server, so you can use existing libraries for OAuth, Auth0, or any other protocol. This repository contains an example of using Auth0 to handle browser-based auth to a Codec Server.

To enable authorization from the Web UI (for both a self-hosted Temporal Service and Temporal Cloud), your Codec Server must be an HTTPS Server.

Temporal Cloud

The Temporal Cloud UI provides an option to pass access tokens (JWT) to your Codec Serv
payl

You
inte
adds
you

In yo
authorization header) against our JWKS endpoint.

COOKIE POLICY

We use necessary cookies to make our site work. With your permission, we also use optional cookies to analyze site usage and assist in our marketing efforts. Read more in our cookie policy.

Do not sell or share my personal information.

The token provided from Temporal Cloud UI contains the email identifier of the person requesting access to the payloads. Based on the permissions you have provided to the user in your access control systems, set conditions in your Codec Server whether to return decoded payloads or just return the original encoded payloads.

Self-hosted Temporal Service

On a self-hosted Temporal Service, configure authorization in the Web UI configuration in your Temporal Service setup.

With this enabled, you can pass access tokens to your Codec Server and validate the requests from the Web UI to the Codec Server endpoints that you set. Note that with a self-hosted Temporal Service, you must explicitly configure authorization specifications for the Web UI and CLI.

Sample request/response

Consider the following sample request/response when creating and hosting a Codec Server with the following specifications:

- Scheme: `https`
- Host: `dev.mydomain.com/codec`
- Path: `/decode`

```
HTTP/1.1 POST /decode
Host: https://dev.mydomain.com/codec
Content-Type: application/json
X-Namespace: myapp-dev.acctid123
Au
{"
{"
20
Co
{
```

COOKIE POLICY

We use necessary cookies to make our site work. With your permission, we also use optional cookies to analyze site usage and assist in our marketing efforts. Read more in our cookie policy.

Do not sell or share my personal information.

```
      "messageType": "temporal_shop.orchestrations.v1.StartShoppingCartReque
    },
    "data":{
      "cartId":"example-cart",
      "shopperId":"your-shopper-id-example",
      "email":"your-email@domain.com"
    }}]
}
```

You can also perform remote encoding on an `/encode` endpoint, which looks the same in reverse:

- Scheme: `https`
- Host: `dev.mydomain.com/codec`
- Path: `/encode`

```
HTTP/1.1 POST /encode
Host: https://dev.mydomain.com/codec
Content-Type: application/json
X-Namespace: myapp-dev.acctid123
Authorization: Bearer <token>

{"payloads":[{"metadata":
{"encoding":"json/protobuf","messageType":"temporal_shop.orchestrations.v1.S
cart","shopperId":"your-shopper-id-example","email":"your-email@domain.com"}

200 OK
Content-Type: application/json

{
"e

}
```

COOKIE POLICY

We use necessary cookies to make our site work. With your permission, we also use optional cookies to analyze site usage and assist in our marketing efforts. Read more in our cookie policy.

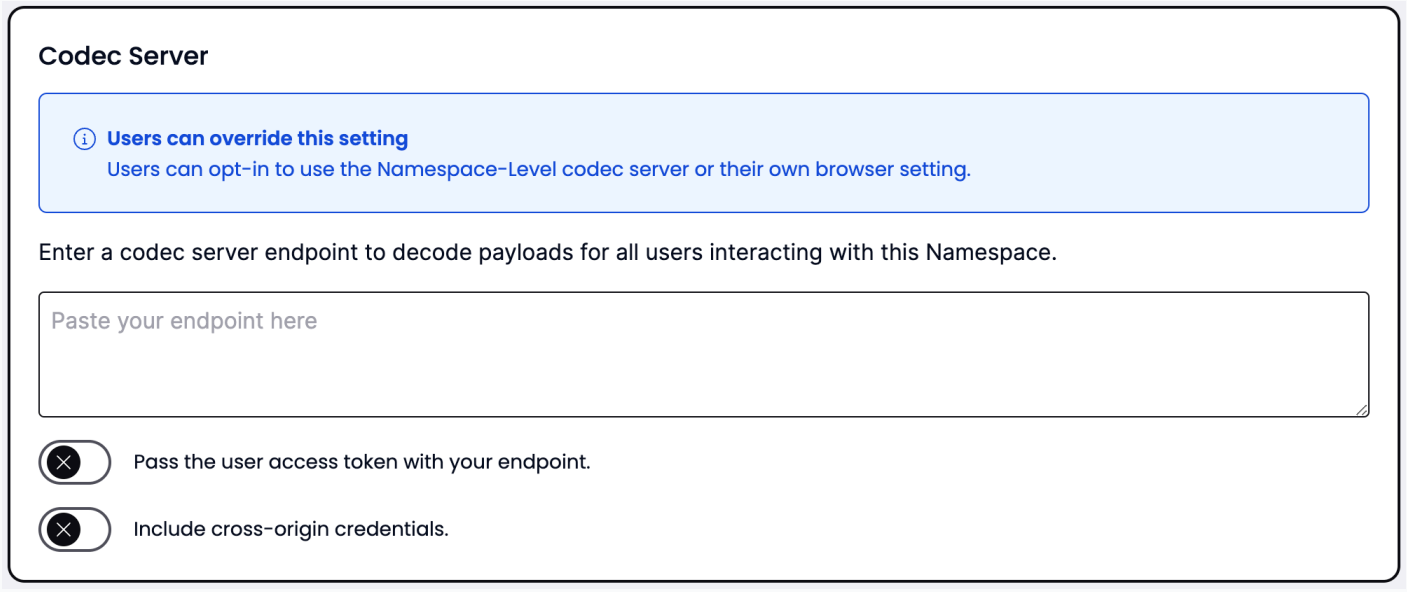
Do not sell or share my personal information.

Set your Codec Server endpoints with Web UI and CLI

After you create your Codec Server and expose the requisite endpoints, set the endpoints in your Web UI and CLI.

Web UI

On Temporal Cloud and self-hosted Temporal Service, you can configure a Codec Server endpoint to be used for a Namespace in the Web UI.



Codec Server endpoint Namespace setting

CAUTION

If your Codec Server is on a private network, your browser may block Temporal Web UI from accessing it. On Chrome, your browser may prompt you to allow access to the Codec Server endpoint. Make sure to allow access to the Codec Server

er

Lo

If

Ch

se

ch

COOKIE POLICY

We use necessary cookies to make our site work. With your permission, we also use optional cookies to analyze site usage and assist in our marketing efforts. Read more in our cookie policy.

Do not sell or share my personal information.

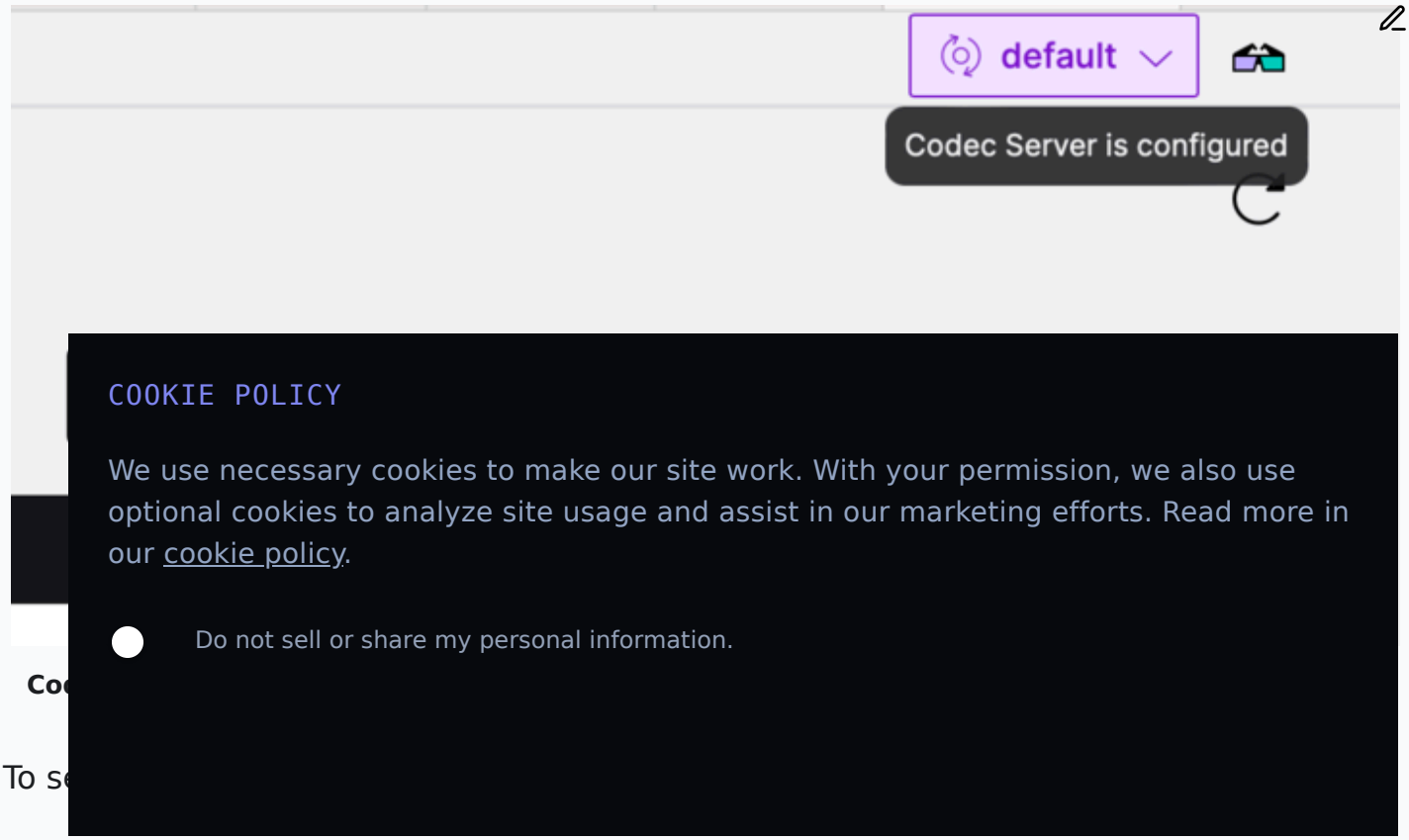
To s

1. In the Web UI, go to Namespaces, select the Namespace where you want to configure the Codec Server endpoint, and click **Edit**.
2. In the **Codec Server** section on the Namespace configuration page, enter your Codec Server endpoint and port number.
3. Optional: If your Codec Server is configured to authenticate requests from Temporal Web UI, enable **Pass access token** to send a JWT access token with the HTTPS requests.
4. Optional: If your Codec Server is configured to verify origins of requests, enable **Include cross-origin credentials**.

On Temporal Cloud, you must have Namespace Admin privileges to add a Codec Server endpoint on the Namespace. Setting a Codec Server endpoint on a Cloud Namespace enables it for all users on the Namespace.

Setting a Codec Server endpoint on a self-hosted Temporal Service enables it for the entire Temporal Service. You can use a single Codec Server to handle different encoding and decoding routes for each namespace.

You can also override the global Codec Server setting at the browser level. This can be useful when developing, testing, or troubleshooting encoding functionality.



Co

To s

1. Navigate to **Workflows** in your Namespace.
2. In the top-right corner, select **Configure Codec Server**.
3. Select whether you want to use the Namespace-level (or Temporal Service-level for self-hosted Temporal Service) or the browser-level Codec Endpoint setting as the default for your browser. In Temporal Cloud:
   - **Use Namespace-level settings, where available. Otherwise, use my browser setting.** Uses the Namespace-level Codec Server endpoint by default. If no endpoint is set on the Namespace, your browser setting is applied.
   - **Use my browser setting and ignore Namespace-level setting.** Applies your browser-level setting by default, overriding the Namespace-level Codec Server endpoint.
4. Enter your Codec Server endpoint and port number.
5. Optional: If your Codec Server is configured to authenticate requests from Temporal Web UI, enable **Pass access token** to send a JWT access token with the HTTPS requests.
6. Optional: If your Codec Server is configured to verify origins of requests, enable **Include cross-origin credentials**.

In a self-hosted Temporal Service with dedicated UI Server configuration, you can also set the codec endpoint in the UI server configuration file:

```
codec:
    endpoint: {{ default .Env.TEMPORAL_CODEC_ENDPOINT "{namespace}"}}
```

CLI

You
endp

For
use

```
te
```

If yo
with each command. For example, to see the decoded output of the Workflow Execution

COOKIE POLICY

We use necessary cookies to make our site work. With your permission, we also use optional cookies to analyze site usage and assist in our marketing efforts. Read more in our cookie policy.

Do not sell or share my personal information.

"yourWorkflow" in the Namespace "yourNamespace", run:

```
temporal --codec-endpoint "http://localhost:8888" --namespace
"yourNamespace" workflow show --workflow-id "yourWorkflow"  --run-id "
<yourRunId>" --output "table"
```

For details, see the CLI reference.

If your Codec Server requires authentication, the Temporal CLI will also accept a `--codec-auth` parameter to supply an authorization header:

```
temporal workflow show \
   --workflow-id converters_workflowID \
   --codec-endpoint 'http://localhost:8081/{namespace}' \
   --codec-auth 'auth-header'
```

Working with large payloads

If your payloads exceed the Temporal Service's size limits, use External Storage to offload large payloads to an external store like Amazon S3. The Web UI displays a claim reference instead of the payload content for externally stored payloads. When External Storage is configured on your Codec Server, the Codec Server can retrieve and decode these payloads for viewing in the Web UI and CLI. See Codec Server with External Storage for details.

Temporal Nexus

The
sent
use

COOKIE POLICY

We use necessary cookies to make our site work. With your permission, we also use optional cookies to analyze site usage and assist in our marketing efforts. Read more in our cookie policy.

Do not sell or share my personal information.

See

Tags