

FastifyViewOptions

August 18, 2026

FastifyViewOptions

Kind: Interface

Source: [packages/platform-fastify/interfaces/external/fastify-view-options.interface.ts](#)

Part of: Platform Fastify

"fastify/view" interfaces

`FastifyViewOptions` defines the configuration accepted by the Fastify view integration. It selects a template engine, identifies template locations, and configures rendering behavior such as layouts, caching, default context, and production settings.

Properties

Property	Type
<code>engine</code>	<code>{ ejs?: any; eta?: any; nunjucks?: any; pug?: any; handlebars?: any; mustache?: any; 'art-template'?: any; twig?: any; liquid?: any; dot?: any; }</code>
<code>templates</code>	<code>`string</code>
<code>includeViewExtension</code>	<code>boolean</code>
<code>options</code>	<code>object</code>
<code>charset</code>	<code>string</code>
<code>maxCache</code>	<code>number</code>
<code>production</code>	<code>boolean</code>
<code>defaultContext</code>	<code>object</code>
<code>layout</code>	<code>string</code>
<code>root</code>	<code>string</code>
<code>viewExt</code>	<code>string</code>

Property	Type
propertyName	string
asyncPropertyname	string

Diagram

```
graph LR
  A[FastifyViewOptions] --> B[engine]
  A --> C[templates / root]
  A --> D[Rendering options]
  A --> E[Cache and production settings]

  B --> B1[EJS]
  B --> B2[Eta]
  B --> B3[Nunjucks]
  B --> B4[Pug]
  B --> B5[Handlebars]
  B --> B6[Other supported engines]

  C --> C1[Template files or directories]
  D --> D1[layout]
  D --> D2[defaultContext]
  D --> D3[includeViewExtension]
  E --> E1[maxCache]
  E --> E2[production]
  E --> E3[charset]
```

Usage

```
import fastify from 'fastify';
import fastifyView from '@fastify/view';
import ejs from 'ejs';
import type { FastifyViewOptions } from '../interfaces/external/fastify-view-options.interface';

const app = fastify();

const viewOptions: FastifyViewOptions = {
  engine: {
    ejs,
  },
  root: `${process.cwd()}/views`,
  templates: `${process.cwd()}/views`,
  layout: 'layouts/main.ejs',
  includeViewExtension: true,
  production: process.env.NODE_ENV === 'production',
  maxCache: 100,
```

```

    charset: 'utf-8',
    defaultContext: {
      applicationName: 'My Fastify App',
    },
    options: {},
  };

  app.register(fastifyView, viewOptions);

  app.get('/', async (_request, reply) => {
    return reply.view('index.ejs', {
      title: 'Home',
      message: 'Welcome!',
    });
  });
});

```

AI Coding Instructions

- Configure exactly one or more compatible engines under `engine` ; ensure the corresponding template package is installed and imported.
- Use `root` and/or `templates` to point to valid template directories or files accessible at runtime.
- Keep `production` aligned with the deployment environment so template caching and reload behavior work as expected.
- Use `defaultContext` only for shared template data; pass request-specific values through `reply.view()` instead.
- When enabling `includeViewExtension` , consistently include template file extensions in view names and layout paths.