

MiddlewareResolver

August 18, 2026

MiddlewareResolver

Kind: Class

Source: `packages/core/middleware/resolver.ts`

Part of: `Core`

`MiddlewareResolver` resolves configured middleware definitions into runtime middleware instances. It centralizes middleware instantiation so the application pipeline can execute middleware consistently and with the appropriate dependencies.

Methods

Method	Signature	Returns
<code>resolveInstances</code>	<code>resolveInstances(moduleRef: Module, moduleName: string)</code>	<code>void</code>

Diagram

```
graph LR
  A[Middleware Configuration] --> B[MiddlewareResolver]
  B --> C[Resolve Middleware Instances]
  C --> D[Application Middleware Pipeline]
  D --> E[Request / Response Handling]
```

Usage

```
import { MiddlewareResolver } from "@your-package/core";

// In most applications, the resolver is created or provided by the core runtime.
function configureMiddleware(resolver: MiddlewareResolver) {
  const middlewareInstances = resolver.resolveInstances();

  // Register the resolved middleware with the application's request pipeline.
  middlewareInstances.forEach((middleware) => {
```

```
    app.use(middleware);  
  });  
}
```

AI Coding Instructions

- Use `resolveInstances()` as the single integration point for converting configured middleware into executable runtime instances.
- Keep middleware registration order intact; middleware behavior often depends on its position in the request pipeline.
- Ensure middleware dependencies are registered in the application's container before resolving instances.
- Avoid manually instantiating middleware when the resolver is available, as this can bypass dependency injection or lifecycle handling.