

ApplicationConfig

August 19, 2026

ApplicationConfig

Kind: Class**Source:** `packages/core/application-config.ts`**Part of:** [Core](#)

`ApplicationConfig` stores application-wide runtime configuration used by the NestJS application bootstrap process. It manages global HTTP prefix settings, WebSocket adapters, pipes, and exception filters so these concerns can be applied consistently across the application.

Methods

Method	Signature	Returns
<code>setGlobalPrefix</code>	<code>setGlobalPrefix(prefix: string)</code>	<code>void</code>
<code>getGlobalPrefix</code>	<code>getGlobalPrefix()</code>	<code>void</code>
<code>setGlobalPrefixOptions</code>	<code>setGlobalPrefixOptions(options: GlobalPrefixOptions<ExcludeRouteMetadata>)</code>	<code>void</code>
<code>getGlobalPrefixOptions</code>	<code>getGlobalPrefixOptions()</code>	<code>GlobalPrefixOptions<ExcludeRouteMetadata></code>
<code>setIoAdapter</code>	<code>setIoAdapter(ioAdapter: WebSocketAdapter)</code>	<code>void</code>
<code>getIoAdapter</code>	<code>getIoAdapter()</code>	<code>WebSocketAdapter</code>
<code>addGlobalPipe</code>	<code>addGlobalPipe(pipe: PipeTransform<any>)</code>	<code>void</code>
<code>useGlobalPipes</code>	<code>useGlobalPipes(pipes: PipeTransform<any>[])</code>	<code>void</code>
<code>getGlobalFilters</code>	<code>getGlobalFilters()</code>	<code>ExceptionFilter[]</code>
<code>addGlobalFilter</code>	<code>addGlobalFilter(filter: ExceptionFilter)</code>	<code>void</code>
<code>useGlobalFilters</code>	<code>useGlobalFilters(filters: ExceptionFilter[])</code>	<code>void</code>
<code>getGlobalPipes</code>	<code>getGlobalPipes()</code>	<code>PipeTransform<any>[]</code>
<code>getGlobalInterceptors</code>	<code>getGlobalInterceptors()</code>	<code>NestInterceptor[]</code>
<code>addGlobalInterceptor</code>	<code>addGlobalInterceptor(interceptor: NestInterceptor)</code>	<code>void</code>
<code>useGlobalInterceptors</code>	<code>useGlobalInterceptors(interceptors: NestInterceptor[])</code>	<code>void</code>
<code>getGlobalGuards</code>	<code>getGlobalGuards()</code>	<code>CanActivate[]</code>
<code>addGlobalGuard</code>	<code>addGlobalGuard(guard: CanActivate)</code>	<code>void</code>

Method	Signature	Returns
<code>useGlobalGuards</code>	<code>useGlobalGuards(guards: CanActivate[])</code>	<code>void</code>
<code>addGlobalRequestInterceptor</code>	<code>addGlobalRequestInterceptor(wrapper: InstanceWrapper<NestInterceptor>)</code>	<code>void</code>
<code>getGlobalRequestInterceptors</code>	<code>getGlobalRequestInterceptors()</code>	<code>InstanceWrapper<NestInterceptor>[]</code>
<code>addGlobalRequestPipe</code>	<code>addGlobalRequestPipe(wrapper: InstanceWrapper<PipeTransform>)</code>	<code>void</code>
<code>getGlobalRequestPipes</code>	<code>getGlobalRequestPipes()</code>	<code>InstanceWrapper<PipeTransform>[]</code>
<code>addGlobalRequestFilter</code>	<code>addGlobalRequestFilter(wrapper: InstanceWrapper<ExceptionFilter>)</code>	<code>void</code>
<code>getGlobalRequestFilters</code>	<code>getGlobalRequestFilters()</code>	<code>InstanceWrapper<ExceptionFilter>[]</code>
<code>addGlobalRequestGuard</code>	<code>addGlobalRequestGuard(wrapper: InstanceWrapper<CanActivate>)</code>	<code>void</code>
<code>getGlobalRequestGuards</code>	<code>getGlobalRequestGuards()</code>	<code>InstanceWrapper<CanActivate>[]</code>
<code>enableVersioning</code>	<code>enableVersioning(options: VersioningOptions)</code>	<code>void</code>
<code>getVersioning</code>	<code>getVersioning()</code>	<code>VersioningOptions</code>

Diagram

```
graph LR
  App[Nest Application] --> Config[ApplicationConfig]

  Config --> Prefix[Global Prefix]
  Config --> PrefixOptions[Prefix Exclusion Options]
  Config --> IoAdapter[WebSocket Adapter]
  Config --> Pipes[Global Pipes]
  Config --> Filters[Global Exception Filters]

  Prefix --> Routes[HTTP Routes]
  IoAdapter --> Gateways[WebSocket Gateways]
  Pipes --> Requests[Incoming Requests]
  Filters --> Errors[Unhandled Exceptions]
```

Usage

```
import { RequestMethod, ValidationPipe } from '@nestjs/common';
import { ApplicationConfig } from '@nestjs/core/application-config';
import { WsAdapter } from '@nestjs/platform-ws';

const config = new ApplicationConfig();

// Configure HTTP route prefixing.
config.setGlobalPrefix('api');
config.setGlobalPrefixOptions({
  exclude: [{ path: 'health', method: RequestMethod.GET }],
});

// Apply request validation globally.
config.addGlobalPipe(
  new ValidationPipe({
```

```

    transform: true,
    whitelist: true,
  }},
);

// Configure the WebSocket transport adapter.
config.setIoAdapter(new WsAdapter());

// Inspect configured values when integrating with application bootstrap.
console.log(config.getGlobalPrefix()); // "api"
console.log(config.getIoAdapter());
console.log(config.getGlobalFilters());

```

AI Coding Instructions

- Use `setGlobalPrefix()` and `setGlobalPrefixOptions()` together when routes need a shared prefix with explicit exclusions.
- Prefer `useGlobalPipes(...pipes)` when registering multiple pipes at once; use `addGlobalPipe()` for incremental registration.
- Ensure global pipes and filters are registered during application setup, before handling requests or initializing gateways.
- Provide a valid `WebSocketAdapter` implementation through `setIoAdapter()` before starting WebSocket gateway infrastructure.
- Treat values returned by getters as application-wide configuration state; avoid mutating returned collections directly.

Relationships

- IMPORTS → `CanActivate`
- IMPORTS → `ExceptionHandler`
- IMPORTS → `NestInterceptor`
- IMPORTS → `PipeTransform`
- IMPORTS → `VersioningOptions`
- IMPORTS → `WebSocketAdapter`
- IMPORTS → `GlobalPrefixOptions`

Used by

8 references from 8 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (8)

- `ExceptionFiltersContext` — `packages/microservices/context/exception-filters-context.ts` :16
- `MicroservicesModule` — `packages/microservices/microservices-module.ts` :23
- `NestMicroservice` — `packages/microservices/nest-microservice.ts` :35
- `TestingModuleOptions` — `packages/testing/testing-module.builder.ts` :29
- `TestingModule` — `packages/testing/testing-module.ts` :26
- `SocketModule` — `packages/websockets/socket-module.ts` :27

- `SocketServerProvider` — `packages/websockets/socket-server-provider.ts` :8
- `WebSocketsController` — `packages/websockets/web-sockets-controller.ts` :29