

# Intercepting HTTP Requests



**Zachary Bennett**

Lead Software Developer

@z\_bennett\_ | <https://www.linkedin.com/in/zbennett10/>

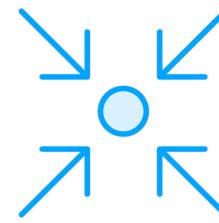
# Module Introduction



**HTTPInterceptor / HttpInterceptorFn**  
**Intercepting requests and responses**  
**Using context tokens with interceptors**  
**Caching requests**



# Interceptor Use Cases



**Help consolidate shared functionality that is used for every request**



**Append default headers to requests**



**Logging requests and responses**



**Optimization via request caching**



**Interceptors help you cut  
down on code duplication  
related to HTTP requests**



## Demo



### Intercepting requests:

- Create your own interceptor class
- Convert this to an interceptor function
- Log all requests
- Registering interceptors



## Demo



### Intercepting responses:

- Update the “LoggingInterceptor”
- Log responses and events by chaining on to the “next” object’s “handle” method



## Demo



### Context tokens and request metadata:

- Implement an interceptor that retries requests
- Read and write context token metadata
- Use context tokens to determine how many more times to retry



## Demo



### Caching requests:

- Implement an interceptor that caches JSON requests

