

1. Blazor Stack 2

1.1 Blazor Stack - Error Handling 2

1.2 Blazor Stack - Obvyklé chyby (FAQ) 2

Blazor Stack

- [Blazor Stack - Error Handling](#)
- [Blazor Stack - Obvyklé chyby \(FAQ\)](#)

Blazor Stack - Error Handling

`OperationFailedException` from server facade -> `Messenger.AddError()`

To pass a non-critical failure from server to client and get it displayed in `HxMessenger`, throw `OperationFailedException` from a facade (or underlying services).

Please note: Although the `exception.ErrorMessage` is displayed as `Messenger.AddError()`, the exception is still re-thrown and you get the yellow-strip-of-death in Blazor Client.

If you don't want to get yellow-strip-of-death in Blazor Client, catch the `OperationFailedException` and act as you need (sometimes you just NOOP).

ApplicationInsights

By default (project template) for Blazor UI (WASM) we are tracking:

- `pageViews`
- `ILogger` events where `LogLevel = Error | Critical`

On server-side (API) there is a "regular" `ApplicationInsights` tracking.

- `OperationFailedException` is filtered-out from tracking.

Blazor Stack - Obvyklé chyby (FAQ)

- [Při volání Facades přes gRPC](#)
 - `NotSupportedException: Specified method is not supported`
 - `RpcException: Status(StatusCode="Unknown", Detail="Exception was thrown by handler.")`

Při volání Facades přes gRPC

NotSupportedException: Specified method is not supported

Chyba nastává na straně gRPC klienta (`Web.Client`), pokud se ze všech možných důvodů nepodaří vytvořit nebo namapovat gRPC proxy, např.:

1. Pokud metoda nemá podporovanou signaturu, např. pokud použijeme více než jeden vstupní parametr.
2. Pokud se uplatní assembly trimming a vykostí nám něco, co je potřeba.

`RpcException: Status(StatusCode="Unknown", Detail="Exception was thrown by handler.")`

Chyba nastává na straně gRPC klienta (`Web.Client`), pokud dostane od gRPC serveru (`Web.Server`) odezvu "`grpc-status: 2; grpc-message: Exception was thrown by handler.`", např.

1. Pokud metoda nemá podporovanou návratovou hodnotu, např. pokud použijeme nested-třidu.