

Pre-handler methods

To add route-level middleware support in Hapi application, you can use `pre` property of options object. Pre it's the ordered array of objects. Each function will be called before the main handler method.

The following example will show how two functions `firstFunction` and `secondFunction` will be called before handler method:

```
const firstFunction = function (request, h) {
  console.log('first call');
  return 1;
};

const secondFunction = function (request, h) {
  console.log('second call');
  return 2;
};

server.route({
  method: 'GET',
  path: '/index',
  options: {
    pre: [{ method: firstFunction,
            assign: 'first' },
          { method: secondFunction,
            assign: 'second' }],
    handler: async (req, h) => {
      console.log(req.pre)
      return h.response({ status: 'ok' });
    }
  }
});
```

- ❑ Method - the `function(request, h, [err])`
- ❑ Assign - the label with the value from a method. You can get it by property in `req.pre` object

The `firstFunction` and `secondFunction` will be called in a row. The returned values from these methods will be passed to `req.pre` object:

```
console.log(req.pre)
// first call
// second call
// { first: 1, second: 2 }
```

You can access the previous value from `firstFunction` inside the `secondFunction`:

```
const firstFunction = function (request, h) {
  console.log('first call')
  return 1;
};

const secondFunction = function (request, h) {
  console.log('second call')
  console.log(request.pre);
  // { first: 1 }
  return 2;
};
```