


```
// src/modules/Timeliste.js
```

```
define(function (require) {
```

```
    var TimelisteService = require('services/TimelisteService'),  
        tilViewModel = require('helpers/tilViewModel');
```

```
    return {
```

```
        timer: function (år, måned) {
```

```
            return tilViewModel(TimelisteService.hentTimer(år, måned));
```

```
        }
```

```
    };
```

```
});
```

```
// src/modules/Timeliste.js
define(function (require) {
    var TimelisteService = require('services/TimelisteService'),
        tilViewModel = require('helpers/tilViewModel');

    return {
        timer: function (år, måned) {
            return tilViewModel(TimelisteService.hentTimer(år, måned));
        }
    };
});
```

```
// spec/modules/TimelisteSpec.js
describe('modules/Timeliste', function () {
    var Timeliste, Service, tilViewModel;

    beforeEach(function () {
        Timeliste = require('modules/Timeliste');
        Service = require('services/TimelisteService');
        tilViewModel = require('helpers/tilViewModel');
    });

    it('henter timer fra TimelisteService', function () { /*...*/ });
});
```



```
/**
 * Utfordringer:
 * - tilstand
 * - initialisering
 * - rene funksjoner
 */

// spec/modules/TimelisteSpec.js
describe('modules/Timeliste', function () {
  var Timeliste, Service, tilViewModel;

  beforeEach(function () {
    Timeliste = require('modules/Timeliste');
    Service = require('services/TimelisteService');
    tilViewModel = require('helpers/tilViewModel');
  });

  it('henter timer fra TimelisteService', function () { /*...*/ });
});
```

```
// spec/modules/TimelisteSpec.js
describe('modules/Timeliste', function () {
  var Timeliste, Service, tilViewModel;

  beforeEach(function () {
    Service = require('services/TimelisteService');
    tilViewModel = jasmine.createSpy();

    var requireFacade = function (dep) {
      if (dep === 'helpers/tilViewModel')
        return tilViewModel;
      return require(dep);
    };

    var TimelisteFactory = require('factory!modules/Timeliste');
    Timeliste = TimelisteFactory(requireFacade);
  });

  it('henter timer fra TimelisteService', function () { /*...*/ })
});
```



```
// spec/modules/TimelisteSpec.js
describe('modules/Timeliste', function () {
  var Timeliste, Service, tilViewModel;

  beforeEach(function () { /*...*/ });

  it('henter timer fra TimelisteService', function () {
    // Arrange
    var år = 2013;
    var måned = 9;
    var mockTimer = [ /*...*/ ];
    var mockResultat = [ /*...*/ ];

    spyOn(Service, 'hentTimer').andCallFake(function (arg1, arg2) {
      if (arg1 === år && arg2 === måned)
        return mockTimer;
      return null;
    });

    tilViewModel.andCallFake(function (arg1) {
      if (arg1 === mockTimer) return mockResultat;
      return null;
    });

    // Act
```

```
/**
 * - factory.js
 * - isolate.js
 * - Mockito
 * - Automocker (C#)
 */
```

```
// spec/modules/TimelisteSpec.js
describe('modules/Timeliste', function () {
  var Timeliste, Service, tilViewModel;

  beforeEach(function () {
    Timeliste = autoMock('modules/Timeliste');
    Service = getMockFor('services/TimelisteService');
    tilViewModel = getMockFor('helpers/tilViewModel');
  });

  it('henter timer fra TimelisteService', function () {
    // Arrange
    var år = 2013;
    var måned = 9;
    var mockTimer = [/*...*/];
    var mockResultat = [/*...*/];

    when(Service.hentTimer)
      .isCalledWith(år, måned).thenReturn(mockTimer);

    when(tilViewModel)
      .isCalledWith(mockTimer).thenReturn(mockResultat);

    // Act
    var resultat = Timeliste.timer(år, måned);
```