

Appendix 2 – Custom FullCalendar Four Week View

```
fcViews.FourWeek = FourWeek; // register the view
```

```
function FourWeek(calendar) {  
  BasicView.call(this, calendar); // call the super-constructor  
}
```

```
FourWeek.prototype = createObject(BasicView.prototype); // define the super-class  
$.extend(FourWeek.prototype, {
```

```
  name: 'FourWeek',
```

```
  incrementDate: function(date, delta) {  
    //We want to only move forward or backwards 4 weeks  
    return date.clone().stripTime().add((delta * 4), 'weeks').startOf('week');  
  },
```

```
  render: function(date) {  
    var rowCnt;
```

```
    this.intervalStart = date.clone().add(7, 'days').stripTime().startOf('week');  
    this.intervalEnd = this.intervalStart.clone().add(4, 'weeks');
```

```
    this.start = this.intervalStart.clone();  
    this.start = this.skipHiddenDays(this.start); // move past the first week if no visible days  
    this.start.startOf('week');  
    this.start = this.skipHiddenDays(this.start); // move past the first invisible days of the week
```

```
    this.end = this.intervalEnd.clone();  
    this.end = this.skipHiddenDays(this.end, -1, true); // move in from the last week if no visible  
days  
    this.end.add((7 - this.end.weekday()) % 7, 'days'); // move to end of week if not already  
    this.end = this.skipHiddenDays(this.end, -1, true); // move in from the last invisible days of the  
week
```

```
    rowCnt = Math.ceil( // need to ceil in case there are hidden days  
      this.end.diff(this.start, 'weeks', true) // return float=true  
    );  
    if (this.isFixedWeeks()) {  
      this.end.add(6 - rowCnt, 'weeks');  
      rowCnt = 6;  
    }  
  }
```

```
    this.title = (this.calendar.formatDate(this.intervalStart, this.opt('titleFormat')) + " - " +  
this.calendar.formatDate(this.intervalEnd.clone().subtract(1, 'days'), this.opt('titleFormat')));
```

```
    BasicView.prototype.render.call(this, rowCnt, this.getCellsPerWeek(), true); // call the super-  
method  
  },
```

```
  // Overrides the default BasicView behavior to have special multi-week auto-height logic  
  setGridHeight: function(height, isAuto) {
```

```

isAuto = isAuto || this.opt('weekMode') === 'variable'; // LEGACY: weekMode is deprecated

// if auto, make the height of each row the height that it would be if there were 6 weeks
if (isAuto) {
    height *= this.rowCnt / 6;
}

distributeHeight(this.dayGrid.rowEls, height, !isAuto); // if auto, don't compensate for height-
hogging rows
},

isFixedWeeks: function() {
    var weekMode = this.opt('weekMode'); // LEGACY: weekMode is deprecated
    if (weekMode) {
        return weekMode === 'fixed'; // if any other type of weekMode, assume NOT fixed
    }

    return this.opt('fixedWeekCount');
}

});

;;

```