function paginate(

totalItems: number,

currentPage: number = 1,

pageSize: number = 10,

maxPages: number = 10

) {

// calculate total pages

let totalPages = Math.ceil(totalItems / pageSize);

// ensure current page isn't out of range

if (currentPage < 1) {

currentPage = 1;

} else if (currentPage > totalPages) {

currentPage = totalPages;

}

let startPage: number, endPage: number;

if (totalPages <= maxPages) {

// total pages less than max so show all pages

startPage = 1;

endPage = totalPages;

} else {

// total pages more than max so calculate start and end pages

let maxPagesBeforeCurrentPage = Math.floor(maxPages / 2);

let maxPagesAfterCurrentPage = Math.ceil(maxPages / 2) - 1;

if (currentPage <= maxPagesBeforeCurrentPage) {

// current page near the start

startPage = 1;

endPage = maxPages;

} else if (currentPage + maxPagesAfterCurrentPage >= totalPages) {

// current page near the end

startPage = totalPages - maxPages + 1;

endPage = totalPages;

} else {

// current page somewhere in the middle

startPage = currentPage - maxPagesBeforeCurrentPage;

endPage = currentPage + maxPagesAfterCurrentPage;

}

}

// calculate start and end item indexes

let startIndex = (currentPage - 1) \* pageSize;

let endIndex = Math.min(startIndex + pageSize - 1, totalItems - 1);

// create an array of pages to ng-repeat in the pager control

let pages = Array.from(Array((endPage + 1) - startPage).keys()).map(i => startPage + i);

// return object with all pager properties required by the view

return {

totalItems: totalItems,

currentPage: currentPage,

pageSize: pageSize,

totalPages: totalPages,

startPage: startPage,

endPage: endPage,

startIndex: startIndex,

endIndex: endIndex,

pages: pages

};

}

export = paginate;