www.bytescout.com

How to display PDF in browser and select area using PDF in Javascript and PDF.js v1.x with Multiple ByteScout SDK

How to display PDF in browser and select area using PDF in Javascript and PDF.js v1.x

Learn how to display PDF in browser and select area using PDF in Javascript and PDF.js v1.x with this source code sample. Multiple ByteScout SDK is Bundle of SDK components can be used to implement multistep workflow. For example, multiple SDKs can be used to implement 2-3 steps code like a search for keywords inside documents and highlights of results found or app that splits pdf based on visually detected lines inside documents and you can use it to display PDF in browser and select area using PDF with Javascript and PDF.js v1.x.

Fast application programming interfaces of Multiple ByteScout SDK for Javascript and PDF.js v1.x plus the instruction and the code below will help you quickly learn how to display PDF in browser and select area using PDF. Just copy and paste the code into your Javascript and PDF.js v1.x application's code and follow the instruction. Further enhancement of the code will make it more vigorous.

Our website provides trial version of Multiple ByteScout SDK for free. It also includes documentation and source code samples.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout Robotic Process Automation

Explore API Documentation

Get Free Training for ByteScout Robotic Process Automation

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

code.js

```
// PDF file to display:
var url = 'https://cdn.mozilla.net/pdfjs/tracemonkey.pdf';
// Note: if absolute URL from the remote server is provided,
// you should configure the CORS header on your server.

// The workerSrc property should be specified
PDFJS.workerSrc = 'https://cdnjs.cloudflare.com/ajax/libs/pdf.js/1.10.97/pdf.worker.js';

var pdfDoc = null,
  pageNum = 1,
  pageRendering = false,
  pageNumPending = null,
  scale = 1,
  canvas = document.getElementById('the-canvas'),
  ctx = canvas.getContext('2d');

/**
 * Get page info from document, resize canvas accordingly, and render page.
 * @param num Page number.
 */
function renderPage(num) {
  pageRendering = true;
  // Using promise to fetch the page
  pdfDoc.getPage(num).then(function (page) {
    var viewport = page.getViewport(scale);
    canvas.height = viewport.height;
    canvas.width = viewport.width;

    // Render PDF page into canvas context
    var renderContext = {
      canvasContext: ctx,
      viewport: viewport
    };
    var renderTask = page.render(renderContext);

    // Wait until the end of the rendering
    renderTask.promise.then(function () {
      pageRendering = false;
      if (pageNumPending !== null) {
        // New page rendering is pending
        renderPage(pageNumPending);
        pageNumPending = null;
      }
    });
  });

  // Update page counters
  document.getElementById('page_num').textContent = num;
}

/**
 * If another page rendering in progress, wait until it finished;
 * otherwise, execute the rendering immediately.
 */
function queueRenderPage(num) {
  if (pageRendering) {
    pageNumPending = num;
  } else {
    renderPage(num);
  }
}

/**
 * Displays previous page.
 */
function onPrevPage() {
  if (pageNum <= 1) {
    return;
  }
```

```
  pageNum--;
  displayCoordinates();
  queueRenderPage(pageNum);
}
document.getElementById('prev').addEventListener('click', onPrevPage);

/**
 * Displays next page.
 */
function onNextPage() {
  if (pageNum >= pdfDoc.numPages) {
    return;
  }
  pageNum++;
  displayCoordinates();
  queueRenderPage(pageNum);
}
document.getElementById('next').addEventListener('click', onNextPage);

/**
 * Download PDF asynchronously.
 */
PDFJS.getDocument(url).then(function (pdfDoc_) {
  pdfDoc = pdfDoc_;
  document.getElementById('page_count').textContent = pdfDoc.numPages;

  // Initial/first page rendering
  renderPage(pageNum);
});


// Area selector
var startX,
  startY,
  selectedBoxes = [],
  $selectionMarquee = $('#selectionMarquee'),
  clearSelectedAreasBtn = $('#clearSelectedAreasBtn'),
  wrapper = $("#canvas-wrap"),
  wrapperObj = wrapper[0],
  $allCords = $('#all-cords'),
  positionBox = function ($box, coordinates) {
    $box.css(
      'top', coordinates.top + wrapperObj.offsetTop
    ).css(
      'left', coordinates.left + wrapperObj.offsetLeft
      ).css(
      'height', coordinates.bottom - coordinates.top
      ).css(
      'width', coordinates.right - coordinates.left
      ).css(
      'border', 'solid grey 1px'
      );
  },

  compareNumbers = function (a, b) {
    return a - b;
  },

  getBoxCoordinates = function (startX, startY, endX, endY) {
    var x = [startX, endX].sort(compareNumbers),
      y = [startY, endY].sort(compareNumbers);

    return {
      top: y[0],
      left: x[0],
      right: x[1],
      bottom: y[1]
    };
  },

  trackMouse = function (event) {
    var position = getBoxCoordinates(startX, startY, event.pageX - wrapperObj.offsetLeft, event.pageY - wrapperObj.of
    console.log(position);
    positionBox($selectionMarquee, position);
  },

  displayCoordinates = function () {
    var msg = 'Areas:\n';
```

```
    if (!selectedBoxes.length) {
      $allCords.html('');
      return;
    }
    selectedBoxes.forEach(function (box) {
      msg += '<li>(' + box.left + ', ' + box.top + ') - (' + (box.right) + ', ' + (box.bottom) + ')</li>';
    });
    $allCords.html(msg);
  },

  endMoving = function (event) {
    var position,
      $selectedBox;

    $selectionMarquee.hide();

    position = getBoxCoordinates(startX, startY, event.pageX - wrapperObj.offsetLeft, event.pageY - wrapperObj.offset

    if (position.left !== position.right && position.top !== position.bottom) {
      $selectedBox = $('<div class="selected-box selected-box-item"></div>');
      $selectedBox.hide();
      wrapper.append($selectedBox);

      positionBox($selectedBox, position, wrapper[0]);

      $selectedBox.show();

      selectedBoxes.push(position);
      displayCoordinates();

      wrapper.off('mousemove', trackMouse);
      $selectionMarquee.off('mousemove', trackMouse);
      $selectionMarquee.off('mouseup', endMoving);
    }
  };

wrapper.on('mousedown', function (event) {
  event.preventDefault();
  startX = event.pageX - wrapperObj.offsetLeft;
  startY = event.pageY - wrapperObj.offsetTop;
  positionBox($selectionMarquee, getBoxCoordinates(startX, startY, startX, startY), this);
  $selectionMarquee.show();

  wrapper.on('mousemove', trackMouse);
  $selectionMarquee.on('mousemove', trackMouse);
  $selectionMarquee.on('mouseup', endMoving);
}).on('mouseup', endMoving);

clearSelectedAreasBtn.on('click', function(event){
  selectedBoxes = [];
  $('.selected-box-item').remove();
  displayCoordinates();
});
```

VIDEO

https://www.youtube.com/watch?v=nm_7l0PN1TY

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout Robotic Process Automation Home Page
Explore ByteScout Robotic Process Automation Documentation
Explore Samples
Sign Up for ByteScout Robotic Process Automation Online Training

ON-DEMAND REST WEB API

Get Your API Key
Explore Web API Docs
Explore Web API Samples

visit www.ByteScout.com

visit www.PDF.co