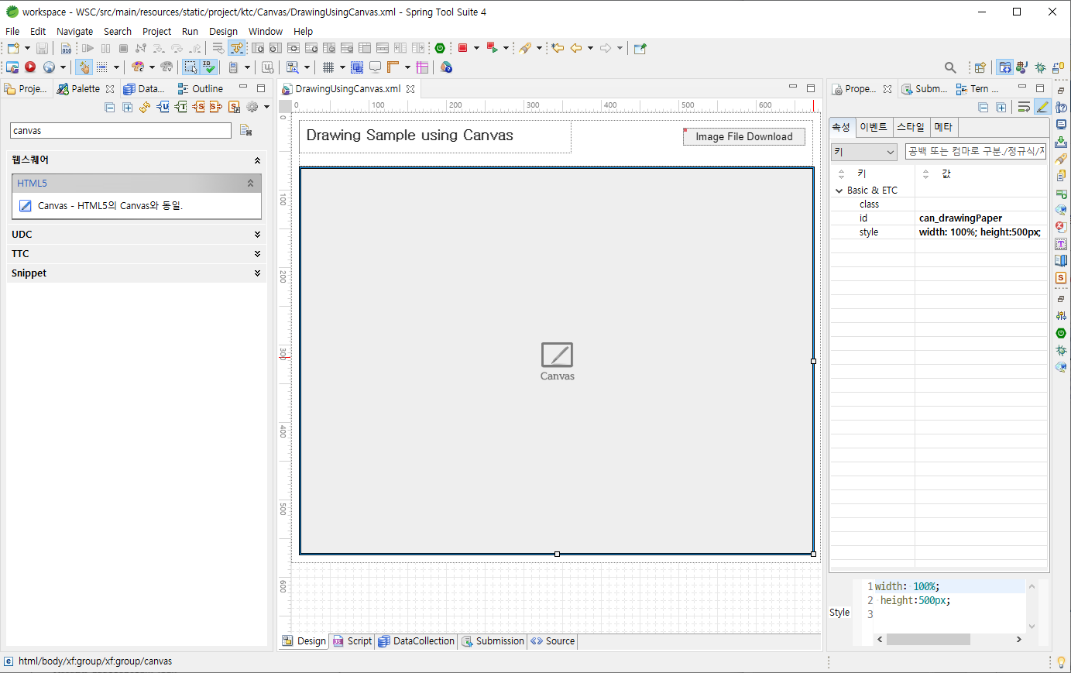
# HTML5 Canvas를 이용한 Drawing 기능 구현하기

WebSquare5 프로젝트를 수행하는 과정에서 그림 그리기 기능이나 전자서명 기능을 필요한 경우가 있습니다. 사용자에게 제공해야 하는 기능이 복잡한 경우에는 별도의 솔루션을 도입하거나 오픈 소스를 활용해서 해당 기능을 구현해야 합니다. 하지만, 간단한 마우스나 펜을 이용한 Drawing 기능을 제공하면 될 경우에는 HTML5 Canvas 태그를 이용해서 간단하게 기능 구현을 할 수 있습니다. 본 문서는 Canvas 태그를 이용해서 웹스퀘어 페이지에서 Drawing 기능을 구현하는 방법을 설명합니다.

## ****1. Canvas를 원하는 위치에 그리기****

Palette View에서 Canvas 컴포넌트를 선택해서 Design View에서 원하는 위치에 Canvas를 배치하고, Canvas 내에 그려진 그림을 파일로 다운로드 받기 위한 'Image File Download' 버튼을 생성합니다.



## 2. Drawing을 기능 구현을 위한 Canvas 및 Event 생성

화면이 로딩될 때 Canvas에 Drawing 기능을 구현하기 위한 초기 설정 및 마우스 이벤트 바인딩 처리를 위한 코드를 다음과 같이 작성합니다.

|  |
| --- |
| // Canvas Element 저장 변수  scwin.canvas = null;  // 2D 방식으로 그림을 그리기 위한 Canvas Context 저장 변수  scwin.ctx = null;  // 현재 좌표 위치 저장 변수  scwin.coord = null;    scwin.onpageload = function() {  scwin.create();  }  /\*\*  \* Image Drawing을 위한 Canvas 및 Event 생성한다.  \*/  scwin.create = function() {  // $p.$() API를 통해서 자신의 Scope 내 Cavans(ID : can\_drawingPaper) element 접근  scwin.canvas = $p.$("#can\_drawingPaper")[0];  // 2D 방식으로 그림을 그리기 위한 Canvas Context를 받음  scwin.ctx = scwin.canvas.getContext("2d");  // Canvas이 초기 좌표  scwin.coord = { x: 0, y: 0 };    // mousedown, mouseup, resize 이벤트 바인딩  document.addEventListener("mousedown", scwin.start);  document.addEventListener("mouseup", scwin.stop);  window.addEventListener("resize", scwin.resize);    scwin.resize();  } |

## 3. Canvas에 Drawing 기능 구현

document에 mousedown, mouseup, mousemove, resize 이벤트의 구현을 통해서 마우스의 클릭과 포인트 이동 시 Canvas 내에 그림이 그려지도록 하기 위한 코드를 다음과 같이 작성합니다.

|  |
| --- |
| /\*\*  \* Mouse Down 이벤트 발생할 때 Move Move 이벤트를 추가해서 Canvas 내에 Drawing 동작을 실행한다.  \*/  scwin.start = function(event) {  document.addEventListener("mousemove", scwin.draw);  scwin.reposition(event);  }  /\*\*  \* 웹 브라우저의 사이즈 변경(Resize 이벤트)이 발생하면 Cavas 사이즈도 변경 시킨다.  \*/  scwin.resize = function() {  scwin.ctx.canvas.width = grp\_drawingPaper.getSize("innerWidth");  scwin.ctx.canvas.height = grp\_drawingPaper.getSize("innerHeight");  }  /\*\*  \* Cavans Drawing할 Mouse 커서의 위치를 설정한다.  \*/  scwin.reposition = function(event) {  scwin.coord.x = event.clientX - scwin.canvas.offsetLeft + document.body.scrollLeft + document.documentElement.scrollLeft;  scwin.coord.y = event.clientY - scwin.canvas.offsetTop + document.body.scrollTop + document.documentElement.scrollTop;  }  /\*\*  \* Mouse Up 이벤트 발생할 때 Move Move 이벤트를 추가해서 Canvas 내에 Drawing 동작을 해제한다.  \*/  scwin.stop = function() {  document.removeEventListener("mousemove", scwin.draw);  }  /\*\*  \* 마우스 Down 상태에서 마우스의 위치를 이동하면 Canvas에 Drawing을 한다.  \*/  scwin.draw = function(event) {  scwin.ctx.beginPath();  scwin.ctx.lineWidth = 2;  scwin.ctx.lineCap = "round";  scwin.ctx.strokeStyle = "gray";  // Canvas에서 펜을 x와 y로 지정된 좌표로 옮김  scwin.ctx.moveTo(scwin.coord.x, scwin.coord.y);  scwin.reposition(event);  // 현재의 드로잉 위치에서 x와 y로 지정된 위치까지 선을 연결함  scwin.ctx.lineTo(scwin.coord.x, scwin.coord.y);  // 윤곽선을 이용하여 도형을 그림  scwin.ctx.stroke();  } |

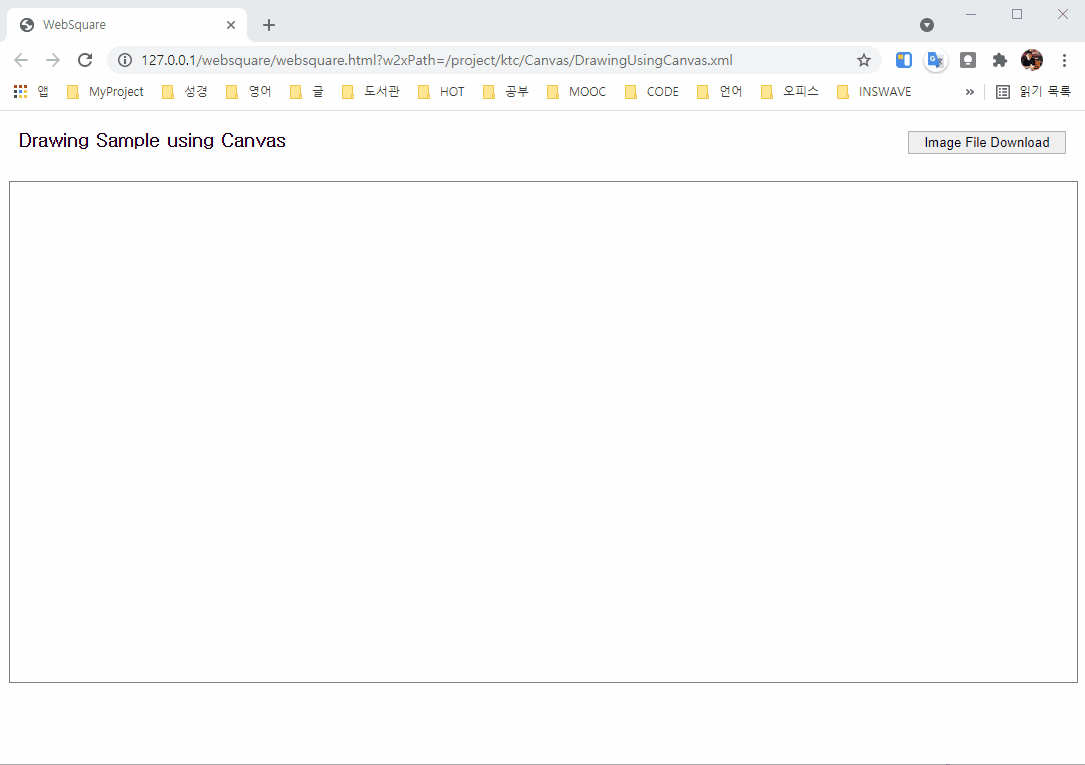
## 4. Drawing 화면을 이미지 파일로 저장하기

'Image File Download' 버튼을 클릭하면 Canvas 위에 그려진 그림을 이미지 파일로 로컬에 다운로드 받기 위한 기능을 다음과 같이 구현합니다.

|  |
| --- |
| scwin.btn\_imageFileDownload\_onclick = function(e) {  scwin.save();  }  /\*\*  \* Canvas에 Drawing한 이미지를 파일로 다운로드 받는다.  \*/  scwin.save = function() {  // Anchor 태그를 이용해서 Canvas에 그려진 이미지를 파일로 다운로드 받음  var link = document.createElement('a');  link.download = 'imageDownload.png';  link.href = scwin.canvas.toDataURL();  link.click();  link.delete;  } |

## 5. Canvas Drawing 샘플 화면 시연

아래의 화면은 위에 작성한 코드를 시연한 화면입니다.



## 6. 샘플 코드

* [DrawingUsingCanvas.xml](file:///C:/confluence/download/attachments/19628298/DrawingUsingCanvas.xml%3fversion=1&modificationDate=1631520725668&api=v2) : Canvas를 이용해서 Drawing 기능을 구현한 샘플 파일
* [MainFrame.xml](file:///C:/confluence/download/attachments/19628298/MainFrame.xml%3fversion=1&modificationDate=1631520725688&api=v2) : DrawingUsingCanvas.xml 파일을 WFrame을 이용해서 2번 로딩한 샘플 파일