

displayTemplateJS.js

```
(function() {
    var overrideCtx = {};
    overrideCtx.Templates = {};

    var self = this;
    self.ProjectAliasControlId = "";
    self.ProjectNameControlId = "";

    fieldValidator = function() {
        fieldValidator.prototype.validate = function(value) {
            var isError = false;
            var errorMessage = "";

            if($get(self.ProjectAliasControlId).value ===
            $get(self.ProjectNameControlId).value)
            {
                isError = true;
                errorMessage = ("Project Alias and Project Name cannot have the same
                value.");
            }

            return new SPCientForms.ControlValidator.ValidateResult(isError,
            errorMessage);
        };
    };

    overrideCtx.Templates.Fields = {
        'ProjectAlias': {
            EditForm: function(ctxCurrent) {
                var ctx =
                SPCientTemplates.Utility.GetFormContextForCurrentField(ctxCurrent);

                var controlId = ctx.fieldName + "_davidmwee_control ";
                self.ProjectAliasControlId = controlId;

                var validators = new SPCientForms.ControlValidator.ValidateSet();
                validators.RegisterValidator(new fieldValidator());

                ctx.registerControlValidator(ctx.fieldName, validators);
                ctx.registerValidatorOnErrorCallback(ctx.fieldName, function(result) {
                    SPFormControl.AppendValidatorErrorMessage(controlId, result);
                });

                ctx.registerGetValueCallback(ctx.fieldName, function() {
                    return $get(controlId).value;
                });

                return String.format('<input type="text" id="{0}" name="{0}" />',
                controlId);
            },
            'ProjectName': {
                EditForm: function(ctxCurrent) {
                    var ctx =
                    SPCientTemplates.Utility.GetFormContextForCurrentField(ctxCurrent);

                    var controlId = ctx.fieldName + "_davidmwee_control ";
                    self.ProjectNameControlId = controlId;

                    var validators = new SPCientForms.ControlValidator.ValidateSet();
                    validators.RegisterValidator(new fieldValidator());

                    ctx.registerControlValidator(ctx.fieldName, validators);
                }
            }
        }
    };
});
```

```
        di spl ayTempl ateJS. js
    ctx. regi sterVal i dati onErrorCal l back(ctx. fi el dName, functi on(errorResul t) {
        SPFormControl _AppendVal i dati onErrorMessage(control Id, errorResul t);
    });

    ctx. regi sterGetVal ueCal l back(ctx. fi el dName, functi on() {
        return $get(control Id). val ue;
    });

    return Stri ng. format(' <i nput type="text" i d="{0}" name="{0}" />' ,
control Id);
    }
}
};

SPCI i entTempl ates. Templ ateManager. Regi sterTempl ateOverri des(overri deCtx);
})();
```