

DhamenPay Widget Customization Guide

Overview

This guide explains how to customize the DhamenPay payment widget for your website. The widget uses iframes for secure payment fields, but you can customize the appearance, add custom elements, and enhance the user experience.

Table of Contents

1. [Basic Setup](#)
 2. [Customizing with CSS](#)
 3. [Using wpwlOptions](#)
 4. [Adding Custom Elements](#)
 5. [Creating Unified Input Layouts](#)
 6. [Adding Brand Logos](#)
 7. [Common Customizations](#)
 8. [Troubleshooting](#)
-

Basic Setup

1. Include Required Scripts

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>DhamenPay</title>

    <!-- Load jQuery (required for customization) -->
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

    <!-- Load DhamenPay Widget -->
    <script src="https://dhamendemo.elm.sa/widget/dhamen-pay.js"></script>

    <meta name="viewport" content="width=device-width, initial-scale=1" />
  </head>
  <body>
    <!-- Payment Widget -->
    <dhamen-pay formstyle="card" invoiceid="YOUR_INVOICE_ID"> </dhamen-pay>
  </body>
</html>
```

Customizing with CSS

Basic Form Styling

```
/* Main form container */
.wpwl-form-card {
  background: #2d2d2d !important;
  border-radius: 24px !important;
  padding: 32px !important;
  max-width: 420px !important;
  margin: auto !important;
  box-shadow: 0 20px 60px rgba(0, 0, 0, 0.5) !important;
}

/* Input fields */
input[type="text"],
input[type="tel"] {
  background: transparent !important;
  border: 1px solid #4a4a4a !important;
  border-radius: 16px !important;
  color: #ffffff !important;
  font-size: 15px !important;
  padding: 18px !important;
}

/* Pay button */
.wpwl-button-pay {
  background: linear-gradient(135deg, #4ade80, #22c55e) !important;
  border: none !important;
  border-radius: 50px !important;
  padding: 18px !important;
  font-weight: 700 !important;
  text-transform: uppercase !important;
  letter-spacing: 1px !important;
}
```

Hide Default Elements

```
/* Hide labels */
.wpwl-label {
  display: none !important;
}

/* Hide brand detection */
.wpwl-control-brand {
  display: none !important;
}
```

Using wpwlOptions

The `wpwlOptions` object allows you to customize widget behavior and add custom functionality.

Basic Configuration

```
var wpwlOptions = {
  style: "logos", // Show payment logos

  onReady: function () {
    console.log("Widget loaded successfully");
    // Your custom code here
  },

  onBeforeSubmitCard: function () {
    // Validate before submission
    return true;
  },
};
```

Available Options

Option	Type	Description
<code>style</code>	string	"logos" to show brand logos
<code>brandDetection</code>	boolean	Auto-detect card brand
<code>brandDetectionType</code>	string	"binlist" or "regex"
<code>locale</code>	string	"en" or "ar" for language
<code>onReady</code>	function	Called when widget loads
<code>onBeforeSubmitCard</code>	function	Called before form submission

Adding Custom Elements

Insert Elements Before Pay Button

```
var wpwlOptions = {
  onReady: function () {
    // Create custom element
    var customDiv = $('<div class="custom-element">Custom Text</div>');

    // Insert before pay button
    customDiv.insertBefore(".wpwl-button-pay");
  },
};
```

Add Custom Fields

```

var wpwlOptions = {
  onReady: function () {
    // Add custom field
    var customField = `
      <div class="wpwl-group">
        <label>Cardholder Email</label>
        <input type="email" name="customer.email" placeholder="email@example.com">
      </div>
    `;

    $(customField).insertBefore(".wpwl-button-pay");
  },
};

```

Creating Unified Input Layouts

Create a modern unified card input container with card number, expiry, and CVV in one row.

HTML Structure

```

var wpwlOptions = {
  onReady: function () {
    // Hide original fields
    $(".wpwl-group-cardNumber, .wpwl-group-expiry, .wpwl-group-cvv").hide();

    // Create unified container
    var unifiedContainer = $(`
      <div class="unified-card-container">
        <svg class="card-icon" viewBox="0 0 24 24">
          <rect x="2" y="5" width="20" height="14" rx="2"/>
          <line x1="2" y1="10" x2="22" y2="10"/>
        </svg>
        <input type="text" class="card-number" placeholder="Card number"
maxlength="19">
        <input type="text" class="expiry" placeholder="MM/YY" maxlength="5">
        <input type="text" class="cvv" placeholder="CVV" maxlength="4">
      </div>
    `);

    // Insert into form
    $(".wpwl-form-card").prepend(unifiedContainer);
  },
};

```

CSS for Unified Layout

```
.unified-card-container {
  display: flex;
  align-items: center;
  background: #3a3a3a;
  border: 1px solid #4a4a4a;
  border-radius: 16px;
  padding: 0 20px;
  height: 56px;
  gap: 12px;
}

.card-icon {
  width: 20px;
  height: 20px;
  opacity: 0.5;
  stroke: #999;
  stroke-width: 2;
  fill: none;
}

.unified-card-container input {
  border: none;
  background: transparent;
  color: #999;
  font-size: 15px;
  outline: none;
}

.card-number {
  flex: 1;
}

.expiry {
  width: 70px;
  text-align: center;
}

.cvv {
  width: 50px;
  text-align: center;
}
```

Add Input Formatting

```
// Card number formatting (spaces every 4 digits)
$(".card-number").on("input", function (e) {
  var value = e.target.value.replace(/\D/g, "");
  var formatted = value.replace(/(\d{4})(?=\d)/g, "$1 ");
  if (formatted.length <= 19) {
    e.target.value = formatted;
  }
});
```

```

    }
  });

  // Expiry formatting (MM/YY)
  $(".expiry").on("input", function (e) {
    var value = e.target.value.replace(/\D/g, "");
    if (value.length >= 2) {
      value = value.substring(0, 2) + "/" + value.substring(2, 4);
    }
    e.target.value = value;
  });

  // CVV (numbers only)
  $(".cvv").on("input", function (e) {
    e.target.value = e.target.value.replace(/\D/g, "");
  });

```

Adding Brand Logos

Display accepted payment methods before the pay button.

Implementation

```

var wpwlOptions = {
  onReady: function () {
    var brands = ["VISA", "MASTER", "MADA"];
    var container = $('<div class="brand-logos"></div>');
    var label = $('<span>WE ACCEPT</span>');

    container.append(label);

    brands.forEach(function (brand) {
      var svg = getBrandSvg(brand);
      if (svg) {
        var wrapper = $('<div class="brand-logo"></div>');
        wrapper.append(svg);
        container.append(wrapper);
      }
    });

    container.insertBefore(".wpwl-button-pay");
  },
};

// SVG Helper Function
function getBrandSvg(brand) {
  switch (brand.toUpperCase()) {
    case "VISA":
      return '<svg width="32" height="20">...</svg>';
    case "MASTER":

```

```
    case "MASTERCARD":  
      return '<svg width="32" height="20">...</svg>';  
    case "MADA":  
      return '<svg width="32" height="20">...</svg>';  
    default:  
      return null;  
  }  
}
```

CSS for Brand Logos

```
.brand-logos {  
  display: flex;  
  align-items: center;  
  gap: 8px;  
  padding: 16px 0;  
  margin-bottom: 8px;  
}  
  
.brand-logos span {  
  color: #888;  
  font-size: 11px;  
  font-weight: 600;  
  letter-spacing: 0.5px;  
  margin-right: 4px;  
}  
  
.brand-logo {  
  height: 24px;  
  display: flex;  
  align-items: center;  
}  
  
.brand-logo svg {  
  height: 20px;  
  border-radius: 4px;  
}
```

Common Customizations

1. Dark Theme

```
body {  
  background: #1a1a1a;  
}  
  
.wpwl-form-card {  
  background: #2d2d2d !important;  
}
```

```
    color: #ffffff !important;
  }

  input {
    background: #3a3a3a !important;
    color: #ffffff !important;
    border: 1px solid #4a4a4a !important;
  }
```

2. Light Theme

```
body {
  background: #f5f5f5;
}

.wpw1-form-card {
  background: #ffffff !important;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1) !important;
}

input {
  background: #f8f8f8 !important;
  border: 1px solid #e0e0e0 !important;
}
```

3. Custom Colors

```
:root {
  --primary-color: #4ade80;
  --secondary-color: #22c55e;
  --background: #2d2d2d;
  --input-bg: #3a3a3a;
  --border-color: #4a4a4a;
}

.wpw1-button-pay {
  background: linear-gradient(135deg, var(--primary-color), var(--secondary-color)) !important;
}
```

4. Responsive Design

```
@media (max-width: 768px) {
  .wpw1-form-card {
    padding: 24px !important;
    border-radius: 16px !important;
  }
```



```
}

input {
  font-size: 14px !important;
  padding: 14px !important;
}
}
```

5. RTL Support (Arabic)

```
[dir="rtl"] .wpwl-form-card {
  direction: rtl;
}

[dir="rtl"] input {
  text-align: right;
}
```

6. Add Loading State

```
var wpwlOptions = {
  onBeforeSubmitCard: function () {
    $(".wpwl-button-pay").text("PROCESSING...");
    $(".wpwl-button-pay").prop("disabled", true);
    return true;
  },
};
```

Troubleshooting

Issue: jQuery is not defined

Solution: Make sure jQuery is loaded before the widget script

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script src="https://dhamendemo.elm.sa/widget/dhamen-pay.js"></script>
```

Issue: Styles not applying

Solution: Use `!important` to override iframe styles

```
.wpwl-form-card {
  background: #2d2d2d !important;
```

```
}
```

Issue: Custom fields not submitting

Solution: Ensure field names follow the correct format

```
<input name="customer.email" /> <input name="customer.phone" />
```

Issue: onReady not firing

Solution: Define wpwlOptions before the widget loads

```
<script>
  var wpwlOptions = { onReady: function() { ... } };
</script>
<script src="https://dhamendemo.elm.sa/widget/dhamen-pay.js"></script>
```

Issue: CSP (Content Security Policy) errors

Solution: Add CSP meta tag (for development only)

```
<meta http-equiv="Content-Security-Policy" content="default-src * 'unsafe-inline'
'unsafe-eval';" />
```

Issue: Payment fields are iframes and can't be styled directly

Solution: Create custom input fields and hide the original iframes

```
$(".wpwl-group-cardNumber").hide();
// Create custom field
```

Best Practices

1. **Always load jQuery first** before the DhamenPay widget
 2. **Use !important** for CSS overrides due to iframe restrictions
 3. **Test on multiple devices** for responsive design
 4. **Validate inputs** before submission using `onBeforeSubmitCard`
 5. **Use setTimeout** if elements aren't ready in `onReady`
 6. **Keep security in mind** - never log sensitive card data
 7. **Test in production mode** after development
-

Example: Complete Custom Implementation

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>DhamenPay Custom</title>

    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script src="https://dhamendemo.elm.sa/widget/dhamen-pay.js"></script>

    <style>
      body {
        font-family: -apple-system, sans-serif;
        background: #1a1a1a;
        display: flex;
        align-items: center;
        justify-content: center;
        min-height: 100vh;
        padding: 20px;
      }

      .wpwl-form-card {
        background: #2d2d2d !important;
        border-radius: 24px !important;
        padding: 32px !important;
        max-width: 420px !important;
        box-shadow: 0 20px 60px rgba(0, 0, 0, 0.5) !important;
      }

      .wpwl-group-cardNumber,
      .wpwl-group-expiry,
      .wpwl-group-cvv,
      .wpwl-label,
      .wpwl-control-brand {
        display: none !important;
      }

      .unified-card {
        display: flex;
        align-items: center;
        background: #3a3a3a;
        border: 1px solid #4a4a4a;
        border-radius: 16px;
        padding: 0 20px;
        height: 56px;
        gap: 12px;
        margin-bottom: 16px;
      }

      .unified-card input {
        border: none;
```

```

        background: transparent;
        color: #999;
        font-size: 15px;
        outline: none;
    }

    .card-number {
        flex: 1;
    }
    .expiry {
        width: 70px;
        text-align: center;
    }
    .cvv {
        width: 50px;
        text-align: center;
    }

    .wpwl-button-pay {
        background: linear-gradient(135deg, #4ade80, #22c55e) !important;
        border: none !important;
        border-radius: 50px !important;
        padding: 18px !important;
        font-weight: 700 !important;
        text-transform: uppercase !important;
        color: white !important;
        width: 100% !important;
    }
</style>
</head>
<body>
    <dhamen-pay formstyle="card" invoiceid="YOUR_INVOICE_ID"></dhamen-pay>

    <script>
        var wpwlOptions = {
            onReady: function () {
                $(".wpwl-group-cardNumber, .wpwl-group-expiry, .wpwl-group-cvv").hide();

                var unified = $(`
                    <div class="unified-card">
                        <input type="text" class="card-number" placeholder="Card number"
maxlength="19">
                        <input type="text" class="expiry" placeholder="MM/YY" maxlength="5">
                        <input type="text" class="cvv" placeholder="CVV" maxlength="4">
                    </div>
                `);

                $(".wpwl-form-card").prepend(unified);

                // Add formatting
                $(".card-number").on("input", function (e) {
                    var v = e.target.value.replace(/\D/g, "");
                    e.target.value = v.replace(/(\d{4})(?=\d)/g, "$1 ");
                });
            }
        };
    </script>

```

```
    },  
    };  
</script>  
</body>  
</html>
```

Additional Resources

- **DhamenPay Documentation:** Contact support for official docs
- **jQuery Documentation:** <https://api.jquery.com/>
- **CSS Tricks:** <https://css-tricks.com/>
- **MDN Web Docs:** <https://developer.mozilla.org/>

Support

For technical support or questions about DhamenPay integration:

- Visit: <https://dhamendemo.elm.sa>
- Email: support@dhamen.sa

Version: 1.0

Last Updated: October 2025

Author: DhamenPay Integration Team