Digioh Regex Validation Extension

Last Modified on 06/24/2026 4:24 pm EDT

What You'll Learn

- **What** the Regex Validation Extension does
- **How** it integrates into your Digioh form
- **Which** metadata keys to configure
- **How** to install and use the extension
- **Best practices** for patterns, messages, and skip logic
- **Troubleshooting** common issues

Overview

This Extension adds **field-level** client-side regex checks to any Digioh form. When a visitor submits the form, it validates each configured field against your JavaScript regular expression.

- If **all** patterns pass, the form submission continues.
- If **any** pattern fails, submission is blocked and your custom error message is returned.

What It Does

1. **Discovers** all fields tagged with regex_to_check metadata
2. **Builds** a config array of { selector, regex, msg, skipValidation, page }
3. **Runs** checks on "After Form Validation"
4. **Returns** your validation_msg on the first failure (blocking submit)
5. **Returns** true if every pattern passes

Metadata Configuration

Where to add metadata: Editor → Page → Form → Fields → Select Field → **Metadata** section

Key	Scope	Description	Example
regex_to_check	**Field**	JavaScript regex (no surrounding slashes), e.g. use ^...$ anchors for full-value match.	^\d{5}$
validation_msg	**Field**	Error text returned on failure. If omitted, defaults to **Invalid value**.	Please enter 5 digits
skip_regex_validation	**Field**	Set to true to bypass checking this field (string 'true' or boolean true).	true

Installation & Usage

1. **Search and install the Extension here**
2. **Publish**

Some Regex Best Practices

- **Anchor your patterns** (^...$) to enforce full-string matches.
- **Escape** any user-input characters (e.g. use \\. for literal dots).
- **Keep messages concise** but clear, e.g. "Enter a valid email" rather than "Oops!".
- **Use skip_regex_validation** sparingly—only when a field should bypass checks under certain conditions.
- **Group related fields** on the same page with consistent naming/selector conventions.
- **Log invalid patterns** at init time—this surfaces misconfigured regex immediately in your console.

Troubleshooting

Issue	Cause	Solution
"Invalid RegExp ..." in console	Malformed pattern in metadata.	Check regex_to_check for missing escapes or slashes.
Field always passes or always fails	Missing anchors or wrong selector.	Verify ^...$ and that selector matches your input.
Error message not showing	validation_msg metadata missing or empty.	Add or update validation_msg on the field.
Skipped validation unexpectedly	skip_regex_validation attribute set to true	Remove or correct the skip flag in field metadata.

Regex with Literals

The extension's loader looks for leading/trailing slashes; you can declare the pattern with /.../, and then escape any internal " or ' with a backslash. For example:

Example: /^[A-Za-z\s'\"]+$/

Test Values "O'Brien" ✅ O'Brien ✅ "Hello World" ✅ "O'Bríen" (curly) ✅

If you prefer not to rely on raw backslashes at all, you can use the Unicode-escape notation for quotes:

Example: /^[A-Za-z\s\u0027\u0022]+$/

Where \u0027 → ' \u0022 → "

These are plain ASCII-range escapes and cannot be stripped or mangled. These both approaches will make sure the pass value survive the Box Editor's sanitation

Pro Tip:

To test your regex quickly, use a website such as:

https://regex101.com/

This ensures your pattern behaves as expected before adding it to Digioh.