

# RDForms: Ein JavaScript-Framework für RDF-Formulare im Web

Dein Name – Bachelorarbeit

September 9, 2025

## Abstract

Die Bearbeitung von RDF-Daten gilt als komplex, da RDF-Graphen für Menschen nur schwer direkt lesbar und editierbar sind. Mit RDForms steht ein JavaScript-Framework zur Verfügung, das RDF-Daten nutzerfreundlich über webbasierte Formulare erfassbar macht. Dieser Artikel beschreibt Hintergrund, Architektur und Einsatzmöglichkeiten von RDForms, vergleicht es mit verwandten Technologien und zeigt Anwendungsfelder für Wissenschaft und Praxis.

## 1 Einführung

Das Resource Description Framework (RDF) ist eine etablierte Spezifikation zur Modellierung strukturierter, verknüpfter Daten im Kontext des Semantic Web [**W3C\_RDF12\_Concepts**]. RDF basiert auf Tripeln (*Subjekt-Prädikat-Objekt*) und ermöglicht es, Informationen maschinenlesbar und interkonnektiert darzustellen. Obwohl RDF eine hohe Ausdrucksmächtigkeit besitzt, stellt die Erfassung und Bearbeitung von RDF-Daten in der Praxis eine Herausforderung dar. Domänenexperten ohne technische RDF-Kenntnisse stoßen häufig an Grenzen, wenn sie direkt mit Serialisierungen wie Turtle oder RDF/XML arbeiten sollen.

An dieser Stelle setzt **RDForms** an: eine JavaScript-Bibliothek, die eine Brücke zwischen RDF-Datenmodellen und benutzerfreundlichen Webformularen schlägt. RDForms erlaubt es, Templates für Formulare zu definieren, die automatisch aus RDF-Graphen befüllt und zurückgeschrieben werden. Damit entsteht eine klare Trennung zwischen Datenmodell und Präsentation, ähnlich wie man es aus klassischen MVC-Architekturen kennt.

## 2 Historischer Hintergrund

Die Ursprünge von RDForms liegen in den frühen 2000er Jahren, als Projekte wie SHAME und RForms erste generische RDF-Editoren bereitstellten [**RDForms\_Overview**]. 2008 wurde RDForms als JavaScript-Lösung

vorgestellt und seit 2011 von der schwedischen Firma MetaSolutions AB weiterentwickelt. Heute bildet RForms einen integralen Bestandteil der Linked-Data-Plattform *EntryScape*, die insbesondere in skandinavischen Ländern in öffentlichen Verwaltungen eingesetzt wird [**EntryScape\_Models**].

### 3 Architektur und Funktionsweise

Die Architektur von RForms besteht aus drei zentralen Komponenten:

- **Templates:** Beschreiben die Struktur des Formulars. Sie geben an, welche RDF-Properties bearbeitet werden sollen, welcher Datentyp erwartet wird und welche Kardinalitäten gelten. Templates werden in JSON beschrieben und können modular wiederverwendet werden.
- **Graph-Handling:** RForms nutzt *rdjson*, um RDF-Graphen im Browser zu verwalten. Ein Graph kann leer initialisiert oder aus externen Quellen (z. B. API, Datei) geladen werden. Änderungen durch den Nutzer werden zurück in den Graph geschrieben.
- **Renderer:** Die Templates werden durch Renderer in eine konkrete UI überführt. RForms stellt Renderer für React, Bootstrap und Material-UI bereit, wodurch sich die Formulare leicht in moderne Frontend-Frameworks integrieren lassen.

Zusätzlich existieren Validator-Komponenten, die sicherstellen, dass Kardinalitäten und Datentypen eingehalten werden. Dies reduziert die Gefahr fehlerhafter RDF-Daten erheblich.

### 4 Beispiel

Das folgende vereinfachte Template beschreibt eine Person mit Vorname, Nachname und Alter:

```
{
  "templates": [
    {
      "id": "foaf::givenName",
      "type": "text",
      "property": "http://xmlns.com/foaf/0.1/givenName",
      "label": {"de": "Vorname"},
      "cardinality": {"min": 1, "max": 1}
    },
    {
      "id": "foaf::familyName",
```

```

    "type": "text",
    "property": "http://xmlns.com/foaf/0.1/familyName",
    "label": {"de": "Nachname"},
    "cardinality": {"min": 1, "max": 1}
  },
  {
    "id": "schema::age",
    "type": "text",
    "property": "http://schema.org/age",
    "datatype": "xsd:integer",
    "label": {"de": "Alter"},
    "cardinality": {"max": 1}
  },
  {
    "id": "personForm",
    "type": "group",
    "items": ["foaf::givenName", "foaf::familyName", "schema::age"]
  }
]
}

```

Dieses Template kann in einer React-Anwendung genutzt werden, um eine Person als RDF-Ressource zu bearbeiten. Der Renderer erzeugt automatisch Eingabefelder, die mit RDF-Tripeln verknüpft sind.

## 5 Vergleich mit Alternativen

Einige verwandte Ansätze sind:

- **XForms**: ein W3C-Standard für deklarative Formulare, jedoch ohne direkte RDF-Integration.
- **SHACL-Form-Engines**: nutzen SHACL-Shapes zur Validierung und Generierung von Formularen, sind aber komplexer in der Einrichtung.
- **DFDP (Declarative Form Description Pipeline)**: ein jüngerer Ansatz, der RDF-Editing unterstützt, allerdings weniger ausgereift ist [Smessaert2024].

Im Vergleich punktet RForms mit einer schlanken JavaScript-Architektur, einfacher Einbettung in bestehende Webanwendungen und einem klaren Fokus auf RDF.

## 6 Einsatzgebiete

RForms wird in verschiedenen Szenarien eingesetzt:

- **Open-Data-Portale:** Formulare für die Beschreibung und Veröffentlichung von Datensätzen.
- **Forschungsdatenmanagement:** Eingabe von Metadaten (Autoren, Publikationen, Projekte).
- **Wissensgraph-Anwendungen:** Bearbeitung von Entitäten (z. B. Personen, Organisationen, Orte).

Durch die Entkopplung von Template und UI eignet sich RDForms besonders für wiederkehrende, standardisierte Eingabeprozesse, bei denen Konsistenz und Validität entscheidend sind.

## 7 Ausblick

Zukünftige Entwicklungen könnten eine stärkere Integration von RDForms mit *Solid Pods* oder anderen verteilten Datenplattformen sein. Auch die automatische Generierung von Templates aus Ontologien oder SHACL-Shapes ist ein spannendes Forschungsthema. Dadurch könnte die manuelle Erstellung von Templates reduziert und die Wiederverwendung bestehender Wissensmodelle gefördert werden.

## 8 Fazit

RDForms stellt eine robuste und praxisnahe Lösung dar, um RDF-Daten über nutzerfreundliche Webformulare zu bearbeiten. Es verbindet deklarative Template-Definition, effizientes Graph-Handling und moderne UI-Integration. Für die Wissenschaft bietet es ein wertvolles Werkzeug, um Linked Data in interaktiven Anwendungen zugänglich zu machen. Im Vergleich zu Alternativen überzeugt RDForms durch Einfachheit, Flexibilität und langjährige Stabilität.