

Das Masterportal ist ein OpenSource-Projekt und basiert als solches auf einer Vielzahl von anderen OpenSource-Bibliotheken. Um die Anzahl der Bibliotheken insgesamt im Projekt klein zu halten und z.B. die Einführung mehrerer Bibliotheken für denselben Zweck zu vermeiden arbeiten wir mit bestimmten Faktoren, nach denen neue Bibliotheken vor einer Integration geprüft und evaluiert werden. Im Vordergrund stehen bei uns u.a. die Bewertung der Stabilität, Nachhaltigkeit, Dokumentation und Community von anderen Projekten um ein möglichst stabiles Produkt zu gewährleisten. Da die Integration einer neuen Bibliothek in den Softwarestack für bestimmte Funktionalitäten erfahrungsgemäß eine Entscheidung für einen längeren Zeitraum bedeutet, sollte dies auch gut abgewogen sein.

Als Entwickler ist hierbei das Vorgehen wie folgt:

1. Prüfen ob der vorhandene Softwarestack (package.json) schon Bibliotheken enthält, mit denen die vorliegende Aufgabe ausgeführt werden kann. Wenn ja muss diese genutzt werden.
2. Wenn zur Bewältigung der Aufgabe keine Bibliothek vorliegt, soll eine kleine Marktrecherche betrieben werden und die Ergebnisse in den folgenden Seiten dokumentiert werden. Das Dokument soll dem PullRequest beigelegt werden.

Es ist zu empfehlen, sich im Vorfeld mit dem Produktmanagement Masterportal kurz abzustimmen um Erfahrungen ggf. schon austauschen zu können.

Bibliothek	Turfjs	Robust-Point-in-polygon	Geolib	
Lizenz	MIT	MIT	MIT	
Anzahl Contributors	145	0	55	
GitHub Stars	8.5k	234	4.1k	
Anzahl Closed Issues im letzten Monat	9	0	0	
Anzahl Active Issues im letzten Monat	17	0	0	
Anzahl von Fragen mit dem entsprechenden Produkt-tag bei stackoverflow	287	0	0	
Größe Package	658 kB	8.7 kB	105 kB	

Hinweis: Hier sollen 3 bis 4 Bibliotheken bezüglich Nachhaltigkeit, Verbreitung, Community etc. verglichen werden.

Vergleich relevanter Funktionen

	Turfjs	Robust-Point-in-polygon	Geolib	
Kann Punkt in einem Polygon ermitteln	ja	ja	ja	

Hinweis: Hier soll der Umfang an Funktionalitäten verglichen werden, v.a. die für den aktuellen Anwendungsfall relevant sind

Bewertungsmatrix: 1-10 Punkte

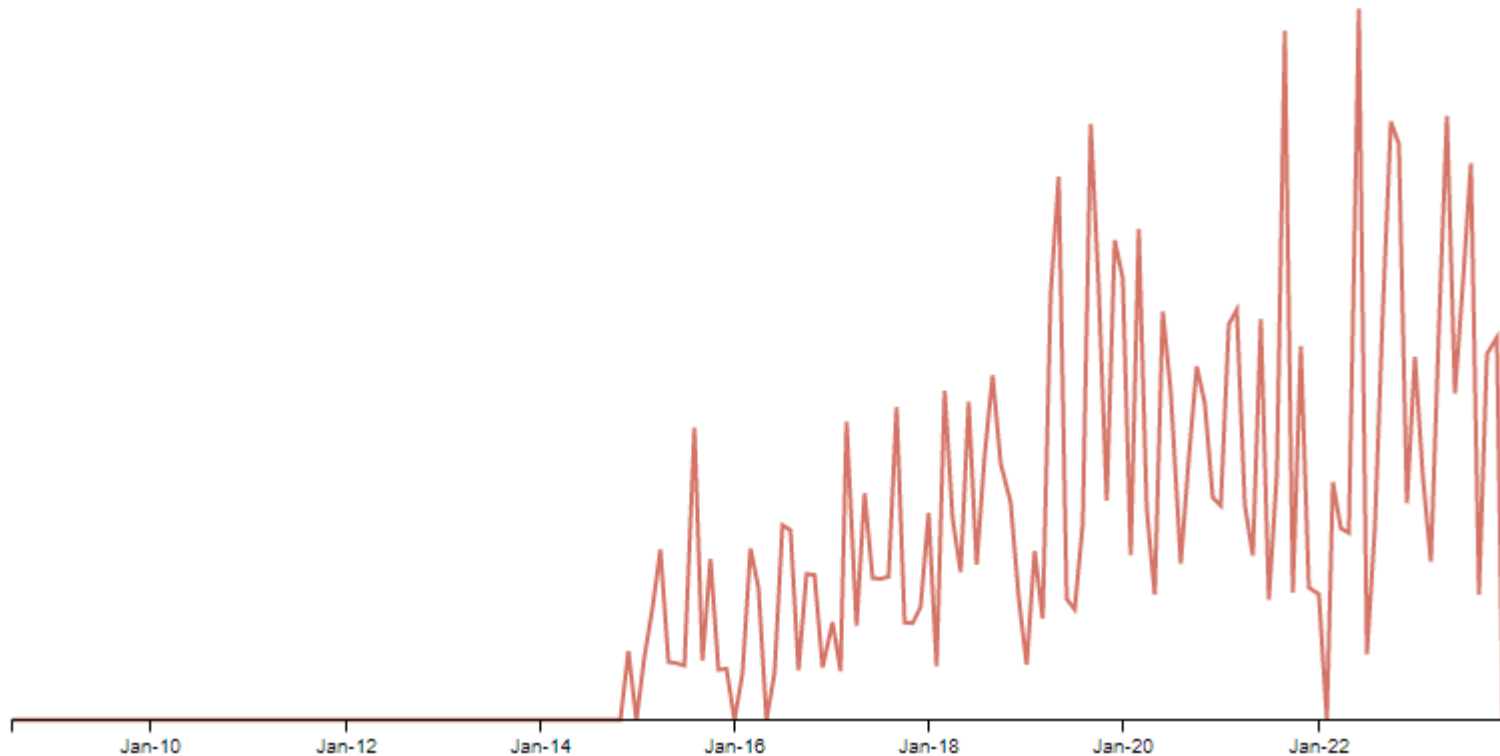
	Turfjs	Robust-Point-in-polygon	Geolib	
Dokumentation und Community	10	4	10	
Syntax einfach	8	2	8	
Nachhaltigkeit	9	3	7	
Universalität und Stabilität	10	5	9	
Summe	37	14	34	

Hinweis: Hier sollen die obigen Kategorien nach eigener subjektiver Einschätzung bewertet werden.

turfjs ×

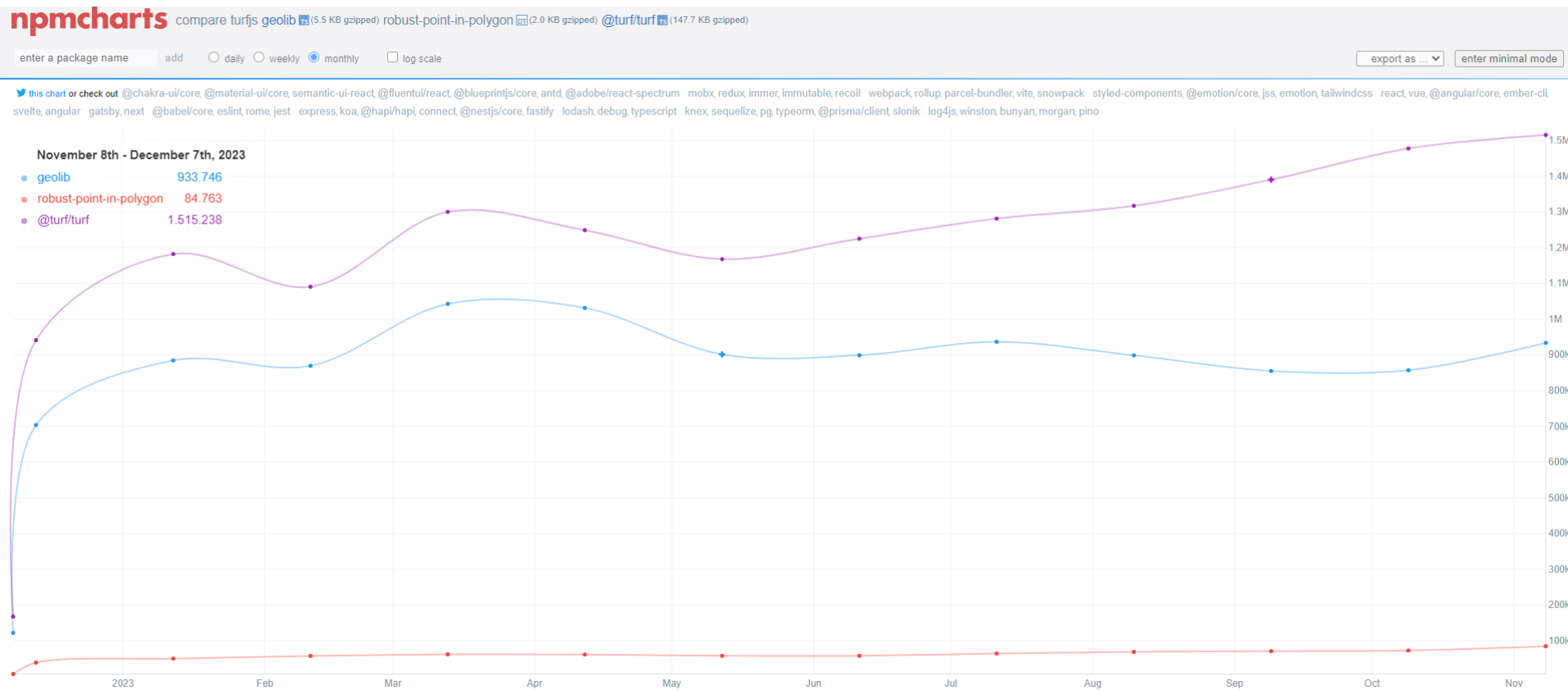
Questions over time ?

Relative: ☒



Hinweis: Hier soll ein Screenshot der Grafik von sotagtrends entsprechend dem obigen Beispiel eingefügt werden. Die Grafik dient als weiterer Indikator, wie groß und aktiv die Nutzercommunity ist.

Popularität: Vergleich NPM (<https://npmcharts.com/>)



Hinweis: Hier soll ein Screenshot der Grafik von npmcharts entsprechend dem obigen Beispiel eingefügt werden. Die Grafik dient als weiterer Indikator, wie groß und aktiv die Nutzercommunity ist.

Hier sollte die Entscheidung für die ausgewählte Bibliothek kurz mit zwei drei Sätzen begründet werden.

Die Entscheidung fällt aus folgenden Gründen auf Turfjs:

- **Aktive Community und Unterstützung:** Turfjs hat eine große und aktive Entwickler-Community, was bedeutet, dass es eine Fülle von Ressourcen, Tutorials und Beispielen gibt, um bei weiterer Entwicklung mit (Multi-)Polygonen zu helfen. Die Community bietet auch regelmäßige Updates und Beiträge zur Verbesserung der Bibliothek.
- **Gut dokumentiert:** Die offizielle Dokumentation von Turfjs ist umfangreich und gut organisiert. Sie bietet eine klare Erläuterung der Funktionalitäten, API-Referenzen und viele Beispiele, die den Einstieg erleichtern und helfen, schnell produktiv zu sein.
- **Turfjs hat eine etwas höhere Bewertung erhalten,** da es seit längerer Zeit eine etablierte und bewährte Bibliothek ist.
- **Zwar bieten auch die anderen Bibliotheken die gleiche Funktionalität,** Turfjs jedoch erscheint am verständlichsten und am Nachhaltigsten auch weiter im GeoSpatial-Anwendungsfall benutzt zu werden.